SCHLOSS DAGSTUHL
Leibniz-Zentrum für Informatik

Dagstuhl Seminar Proceedings 09121

Normative Multi-Agent Systems

Guido Boella, Pablo Noriega, Gabriella Pigozzi ,
Harko Verhagen (Eds.)
Dagstuhl Seminar 09121, 15.03.09 - 20.03.09

ISSN 1862 - 4405

SCHLOSS DAGSTUHL
Leibniz-Zentrum für Informatik

Dagstuhl Seminar Proceedings 09121

Normative Multi-agent Systems

Guido Boella, Pablo Noriega, Gabriella Pigozzi , Harko Verhagen (Eds.)

ISSN 1862 - 4405

Dagstuhl Seminar 09121, 15.03.09 - 20.03.09

Contract Formation through Preemptive Normative Conflict Resolution

Authors: Vasconcelos, Wamberto ; Norman, Timothy J.

Distrust is not Always the Complement of Trust (Position Paper)

Authors: da Costa Pereira, Celia

Dynamic Context Logic and its Application to Norm

Authors: Aucher, Guillaume ; Grossi, Davide ; Herzig, Andreas ; Lorini, Emiliano

Early requirements engineering for e-customs decision support: Assessing overlap in mental models

Authors: Burgemeestre, Brigitte ; Liu, Jianwei ; Hulstijn, Joris ; Tan, Yao-Hua

FSL -- Fibred Security Language

Authors: Genovese, Valerio

How Do Agents Comply with Norms?

Authors: Governatori, Guido ; Rotolo, Antonino

Massively multiple online role playing games as normative multiagent systems

Authors: Magnus, Johansson ; Verhagen, Harko

Modeling and Validating Norms

Authors: Torres da Silva, Viviane ; Braga, Christian

Monitoring Social Expectations in Second Life

Authors: Cranefield, Stephen ; Li, Guannan

Normal = Normative? The Role of Intelligent Agents in Norm Innovation

Authors: Andrighetto, Giulia ; Campennì, Marco ; Cecconi, Federico ; Conte, Rosaria

NorMAS-RE: a Normative Multiagent Approach to Requirements Engineering

Authors: Villata, Serena

Normative Multi-Agent Programs and Their Logics

Authors: Dastani, Mehdi ; Grossi, Davide ; Meyer, John-Jules ; Tinnemeier, Nick

On dissemination mechanism of corporate social responsibility (CSR): Analysis with agent simulation

Authors: Hashimoto, Takashi ; Shinohara, Naoto ; Egashira, Susumu

Partially Observable Markov Decision Processes with Behavioral Norms

Authors: Nickles, Matthias ; Rettinger, Achim

Reflection and Norms: Towards a Model for Dynamic Adaptation for MAS

Authors: Timm, Ingo J. ; Lattner, Andreas D. ; Schumann, Rene

Robust Normative Systems

Authors: Agotnes, Thomas ; van der Hoek, Wiebe ; Wooldridge, Michael

09121 Abstracts Collection
Normative Multi-Agent Systems
— Dagstuhl Seminar —

Guido Boella[1], Pablo Noriega[2], Gabriella Pigozzi[3] and Harko Verhagen[4]

[1] University of Torino, I
guido@di.unito.it
[2] IIIA - CSIC - Barcelona, E
pablo@iiia.csic.es
[3] University of Luxemburg, L
gabriella.pigozzi@uni.lu
[4] Stockholm University, S
verhagen@dsv.su.se

Abstract. From 15.03. to 20.03.2009, the Dagstuhl Seminar 09121 "Normative Multi-Agent Systems " was held in Schloss Dagstuhl – Leibniz Center for Informatics. During the seminar, several participants presented their current research, and ongoing work and open problems were discussed. Abstracts of the presentations given during the seminar as well as abstracts of seminar results and ideas are put together in this paper. The first section describes the seminar topics and goals in general. Links to extended abstracts or full papers are provided, if available.

Keywords. Similarity-based clustering and classification, metric adaptation and kernel design, learning on graphs, spatiotemporal data

Robust Normative Systems

Thomas Agotnes (Bergen University College, NO)

Although normative systems, or social laws, have proved to be a highly influential approach to coordination in multi-agent systems, the issue of *compliance* to such normative systems remains problematic. In all real systems, it is possible that some members of an agent population will not comply with the rules of a normative system, even if it is in their interests to do so. It is therefore important to consider the extent to which a normative system is *robust*, i.e., the extent to which it remains effective even if some agents do not comply with it. We formalise and investigate three different notions of robustness and related decision problems.

We begin by considering sets of agents whose compliance is necessary and/or sufficient to guarantee the effectiveness of a normative system; we then consider quantitative approaches to robustness, where we try to identify the proportion

Dagstuhl Seminar Proceedings 09121
Normative Multi-Agent Systems
http://drops.dagstuhl.de/opus/volltexte/2009/1922

of an agent population that must comply in order to ensure success, and finally, we consider a more general approach, where we characterise the compliance conditions required for success as a logical formula.

Keywords: Normative systems, robustness, fault tolerance, complexity

Joint work of: Agotnes, Thomas; van der Hoek, Wiebe; Wooldridge, Michael

Full Paper: http://drops.dagstuhl.de/opus/volltexte/2009/1897

What do Agent-Based and Equation-Based Modelling Tell us about Social Conventions?

Giulia Andrighetto (ISTC - CNR - Rome, IT)

Ten years ago, during MABS'98, H. Van Dyke Parunak, Robert Savit and Rick L. Riolo discussed the similarities and differences between the Agent-Based Modelling (ABM) and the Equation-Based Modelling (EBM), developing criteria for selecting one or the other approach. The authors concluded that a distinction between them must be made case by case on the basis of practical considerations. In this work we will present and confront some simulation-based and analytical results on the emergence of steady states in a class of coordination games, the congestion games. In particular, our study focuses on the emergence of steady states in traffic-like interactions, drawing on Sen and Airiau's study of the emergence of the precedence rule. We show that, in contrast with Parunak-Savit-Riolo conclusions, in congestion games we should use an integrated approach, mixing the ABM and the EBM frameworks. A crucial feature concerns organization: simulation results are organized in some hierarchical structure since they are generated by our algorithms. For example, in our model, results incorporate peculiar symmetries: i.e. equivalent strategies cannot coexist, while non-equivalent ones can. We endeavor to explicate these symmetric results using EBM.

Keywords: Agent based modelling, conventions, equation based modelling

Joint work of: Andrighetto, Giulia; Cecconi, Federico; Campennì, Marco; Conte, Rosaria

Normal = Normative? The Role of Intelligent Agents in Norm Innovation

Giulia Andrighetto (ISTC - CNR - Rome, IT)

In this paper the results of several agent-based simulations, aiming to test the role of normative beliefs in the emergence and inno- vation of social norms, are presented and discussed.

Rather than mere behavioral regularities, norms are here seen as behaviors spreading to the extent that and because the corresponding commands and beliefs do spread as well. On the grounds of such a view, the present work will endeavour to show that a sudden external constraint (e.g. a barrier preventing agents from moving among social settings) facilitates norm innovation: under such a condition, agents provided with a module for telling what a norm is can generate new (social) norms by forming new normative beliefs, irrespective of the most frequent actions.

Keywords: Norm emergence, agent based simulation

Joint work of: Andrighetto, Giulia; Cecconi, Federico; Campennì, Marco; Conte, Rosaria

Full Paper: http://drops.dagstuhl.de/opus/volltexte/2009/1898

Dynamic Context Logic and its Application to Norm

Guillaume Aucher (University of Luxembourg, LU)

Building on a simple modal logic of context, the paper presents a dynamic logic characterizing operations of contraction and expansion on theories.

We investigate the mathematical properties of the logic, and use it to develop an axiomatic and semantic analysis of norm change in normative systems. The proposed analysis advances the state of the art by providing a formal semantics of norm-change which, at the same time, takes into account several different aspects of the phenomenon, such as permission and obligation dynamics, as well as the dynamics of classificatory rules.

Keywords: Context logic, norm change, deontic logic

Joint work of: Aucher, Guillaume; Grossi, Davide; Herzig, Andreas; Lorini, Emiliano

Full Paper: http://drops.dagstuhl.de/opus/volltexte/2009/1900

A Taxonomy for Ensuring Institutional Compliance in Utility Computing

Tina Balke (Universität Bayreuth, DE)

With the ongoing evolution from closed to open distributed systems and the lifting of the assumption that agents acting in such a system do not pursue own goals and act in the best interest of the society, new problems arise. One of them is that compliance cannot be assumed necessarily and consequently trust issues arise. One way of tackling this problem is by regulating the behavior of the

agents with the help of institutions. However for institutions to function effectively their compliance needs to be ensured. Using a utility computing scenario as sample application, this paper presents a general applicable taxonomy for ensuring compliance that can be consulted for analyzing, comparing and developing enforcement strategies and hopefully will stimulate research in this area.

Keywords: Institutions, Compliance, Enforcement, Regimentation, Norms, Sanctions, Utility Computing

Full Paper: http://drops.dagstuhl.de/opus/volltexte/2009/1901

Normative Systems in Computer Science - Ten Guidelines for Normative Multiagent Systems

Guido Boella (University of Torino, IT)

In this paper we introduce and discuss ten guidelines for the use of normative systems in computer science. We adopt a multiagent systems perspective, because norms are used to coordinate, organize, guide, regulate or control interaction among distributed autonomous systems.

The first six guidelines are derived from the computer science literature.

From the so-called 'normchange' definition of the first workshop on normative multiagent systems in 2005 we derive the guidelines to motivate which definition of normative multiagent system is used, to make explicit why norms are a kind of (soft) constraints deserving special analysis, and to explain why and how norms can be changed at runtime. From the so-called 'mechanism design' definition of the second workshop on normative multiagent systems in 2007 we derive the guidelines to discuss the use and role of norms as a mechanism in a game-theoretic setting, clarify the role of norms in the multiagent system, and to relate the notion of "norm" to the legal, social, or moral literature. The remaining four guidelines follow from the philosophical literature: use norms also to resolve dilemmas, and in general to coordinate, organize, guide, regulate or control interaction among agents, distinguish norms from obligations, prohibitions and permissions, use the deontic paradoxes only to illustrate the normative multiagent system, and consider regulative norms in relation to other kinds of norms and other social-cognitive computer science concepts.

Keywords: Normative systems, Guidelines, Norms, Multiagent systems, Deontic logic

Joint work of: Boella, Guido; Pigozzi, Gabriella; van der Torre, Leendert

Full Paper: http://drops.dagstuhl.de/opus/volltexte/2009/1902

A Framework for Normative MultiAgent Organisations

Olivier Boissier (Ecole des Mines - St. Etienne, FR)

The social and organisational aspects of agency have led to a good amount of theoretical work in terms of formal models and theories. From these different works normative multiagent systems and multiagent organisations are particularily considered in this paper. Embodying such models and theories in the conception and engineering of proper infrastructures that achieve requirements of openness and adaptation, is still an open issue. In this direction, this paper presents and discusses a framework for normative multiagent organisations. Based on the Agents and Artifacts meta-model (A&A), it introduces organisational artifacts as first class entities to instrument the normative organisation for supporting agents activities within it.

Keywords: Normative system, organisation, artifacts, norm enforcement

Joint work of: Boissier, Olivier; Hübner, Jomi Fred

Full Paper: http://drops.dagstuhl.de/opus/volltexte/2009/1903

A Conviviality Measure for Early Requirement Phase

Patrice Caire (University of Luxembourg, L)

In this paper, we consider the design of convivial multi-agent systems. Conviviality has recently been proposed as a social concept to develop multi-agent systems. In this paper we introduce temporal dependence networks to model the evolution of dependence networks and conviviality over time, we introduce epistemic dependence networks to combine the viewpoints of stakeholders, and we introduce normative dependence networks to model the transformation of social dependencies by hiding power relations and social structures to facilitate social interactions. We show how to use these visual languages in design, and we illustrate the design method using an example on virtual children adoptions.

Keywords: Multi-agent systems

Joint work of: Caire, Patrice; van der Torre, Leendert

Full Paper: http://drops.dagstuhl.de/opus/volltexte/2009/1899

A modal logic for reasoning on consistency and completeness of regulations

Laurence Cholvy (ONERA - Toulouse Research Center, FR)

In this paper, we deal with regulations that may exist in multi-agent systems in order to regulate agent behaviour and we discuss two properties of regulations, that is consistency and completeness.

After defining what consistency and completeness mean, we propose a way to consistently complete incomplete regulations. In this contribution, we extend previous works and we consider that regulations are expressed in a first order modal deontic logic.

Keywords: Regulations, consistency, completeness, deontic logic, default logic

Joint work of: Garion, Christophe; Roussel, Stéphanie; Cholvy, Laurence

Full Paper: http://drops.dagstuhl.de/opus/volltexte/2009/1904

A categorization of simulation works on norms

Stephen Cranefield (University of Otago, NZ)

In multi-agent systems, software agents are modelled to possess characteristics and behaviour borrowed from human societies. Norms are expectations of behaviours of the agents in a society. Norms can be established in a society in different ways. In human societies, there are several types of norms such as moral norms, social norms and legal norms (laws). In artificial agent societies, the designers can impose these norms on the agents. Being autonomous, agents might not always follow the norms. Monitoring and controlling mechanisms should be in place to enforce norms. As the agents are autonomous, they themselves can evolve new norms while adapting to changing needs. In order to design and develop robust artificial agent societies, it is important to understand different approaches proposed by researchers by which norms can spread and emerge within agent societies. This paper makes two contributions to the study of norms. Firstly, based on the simulation works on norms, we propose a life-cycle model for norms. Secondly, we discuss different mechanisms used by researchers to study norm creation, spreading, enforcement and emergence.

Keywords: Norms, creation, spreading, enforcement, emergence

Joint work of: Savarimuthu, Bastin Tony Roy; Cranefield, Stephen

Full Paper: http://drops.dagstuhl.de/opus/volltexte/2009/1905

Monitoring Social Expectations in Second Life

Stephen Cranefield (University of Otago, NZ)

Online virtual worlds such as Second Life provide a rich medium for unstructured human interaction in a shared simulated 3D environment. However, many human interactions take place in a structured social context where participants play particular roles and are subject to expectations governing their behaviour, and current virtual worlds do not provide any support for this type of interaction. There is therefore an opportunity to adapt the tools developed in the MAS community for structured social interactions between software agents (inspired by human society) and adapt these for use with the computer-mediated human communication provided by virtual worlds. This paper describes the application of one such tool for use with Second Life. A model checker for online monitoring of social expectations defined in temporal logic has been integrated with Second Life, allowing users to be notified when their expectations of others have been fulfilled or violated. Avatar actions in the virtual world are detected by a script, encoded as propositions and sent to the model checker, along with the social expectation rules to be monitored. Notifications of expectation fulfilment and violation are returned to the script to be displayed to the user. This utility of this tool is reliant on the ability of the Linden scripting language (LSL) to detect events of significance in the application domain, and a discussion is presented on how a range of monitored structured social scenarios could be realised despite the limitations of LSL.

Keywords: Virtual worlds, Second Life, social expectations

Joint work of: Cranefield, Stephen; Li, Guannan

Full Paper: http://drops.dagstuhl.de/opus/volltexte/2009/1906

Normative Multi-Agent Programs and Their Logics

Mehdi Dastani (Utrecht University, NL)

Multi-agent systems are viewed as consisting of individual agents whose behaviors are regulated by an organization artefact. This paper presents a simplified version of a programming language that is designed to implement norm-based artefacts. Such artefacts are specified in terms of norms being enforced by monitoring, regimenting and sanctioning mechanisms. The syntax and operational semantics of the programming language are introduced and discussed. A logic is presented that can be used to specify and verify properties of programs developed in this language.

Keywords: Normative Multi-Agent Systems, Programming Multi-Agent Systems

Joint work of: Dastani, Mehdi; Grossi, Davide; Meyer, John-Jules; Tinnemeier, Nick

Full Paper: http://drops.dagstuhl.de/opus/volltexte/2009/1907

A Meta-model for the Specification of Artificial Institutions using the Event Calculus

Nicoletta Fornara (University of Lugano, CH)

The specification of open interaction systems, which may be dynamically entered and left by autonomous agents, is widely recognized to be a crucial issue in the development of distributed applications on the internet.

The specification of such systems involves two main problems: the first is the definition of a standard way of specifying a communication language for the interacting agents and the context of the interaction; the second, which derives from the assumption of the agents' autonomy, is finding a way to regulate interactions so that agents may have reliable expectations on the future development of the system. A possible approach to solve those problems consists in modeling the interaction systems as a set of artificial institutions. In this chapter we address this issue by formally defining, in the Event Calculus, a repertoire of abstract concepts (like commitment, institutional power, role, norm) that can be used to specify artificial institutions. We then show how, starting from the formal specification of a system and using a suitable tool, it is possible to simulate and monitor the systemŠs evolution through automatic deduction.

Keywords: Artificial Institutions, Open Interaction Systems, Norms, Commitment, Power, Event Calculus

Joint work of: Fornara, Nicoletta; Colombetti, Marco

Designing Ontologies for NMAS: Some Patterns

Aldo Gangemi (ISTC - CNR - Rome, IT)

This paper presents a more comprehensive approach to deal with NMAS ontology specification in a computational environment. Such approach employs semantic web languages such as OWL, RIF, SPARQL, etc., and complies to the eXtreme Design paradigm, which is a method to build an ontology by exploiting user requirements (in the form of competency questions,), and reusable ontology design patterns for both ontology building and evaluation.

The patterns presented here are partly extracted from the ODP community portal, and those that are closely related to the NMAS domain are extracted from the NIC ontology, as well as other, related ones. Some recipes are presented which allow different reasoning styles on NMAS entities (DL classification, subsumption and realization, constructive query answering, rule engines, etc.), and a ranking of the recipes is provided.

The ultimate suggestion is to attach an ontology reasoning component to NMAS, which can be leveraged to perform typical reasoning tasks on the NMAS domain, while leaving NMAS to work on such normalized knowledge, and to concentrate on typical NMAS functionalities, such as dynamics of NMAS worlds.

A proposal to share modelling practices for NMAS on the ODP community portal is also briefly sketched.

Keywords: NMAS ontologies, Ontology design patterns, Collaborative design

FSL – Fibred Security Language

Valerio Genovese (University of Torino, IT)

We develop a fibred security language capable to express statements of the form

$$\{x\}\varphi(x) \textbf{ says } \psi$$

where $\{x\}\varphi(x)$ is the set of all x that satisfy φ and ψ is any formula. φ and ψ may share several free variables.

For example, we can express the following: "A member m of the Program Committee can not accept a paper P_1 in which one of its authors says that he has published a paper with him after 2007"

$$\neg(\{m\}[PC(m) \wedge \{y\}author\_of(y, P_1) \textbf{ says } \exists p(paper(p) \wedge author\_of(m, p) \wedge author\_of(y, p) \wedge year(p) \geq 2007)] \textbf{ says } accept(P_1))$$

Keywords: Access Control, Trust Management, Fibring Logics

Joint work of: Genovese, Valerio; Boella, Guido; Gabbay, Dov M.; van der Torre, Leendert

Full Paper: http://drops.dagstuhl.de/opus/volltexte/2009/1908

How Do Agents Comply with Norms?

Guido Governatori (NICTA Queensland Research Laboratory, AU)

The import of the notion of institution in the design of MASs requires to develop formal and efficient methods for modeling the interaction between agents' behaviour and normative systems. This paper discusses how to check whether agents' behaviour is compliant with the rules regulating them. The key point of our approach is that compliance is a relationship between two sets of specifications: the specifications for executing a process and the specifications regulating it. We propose a logic-based formalism for describing both the semantics of normative specifications and the semantics of compliance checking procedures.

Keywords: Compliance, agents, violations, norms

Joint work of: Governatori, Guido; Rotolo, Antonino

Full Paper: http://drops.dagstuhl.de/opus/volltexte/2009/1909

A note on brute vs. institutional facts

Davide Grossi (University of Amsterdam, NL)

The paper investigates the famous Searlean distinction between "brute" and "institutional" concepts from a logical point of view.

We show how the partitioning of the non-logical alphabet-e.g., into "brute" and "institutional" atoms-gives rise to interesting modal properties. A modal logic, called UpTo-logic, is introduced and investigated which formalizes the notion of (propositional) logical equivalence up to a given signature.

Keywords: Modal logic, brute and institutional facts

Full Paper: http://drops.dagstuhl.de/opus/volltexte/2009/1910

On dissemination mechanism of corporate social responsibility (CSR): Analysis with agent simulation

Takashi Hashimoto (JAIST - Ishikawa, JP)

Corporate Social Responsibility (CSR), such as pro-environmental behaviour and fair trade, is a kind of normative behaviour by private companies to provide a quasi-public good.We study dissemination mechanism of CSR with a multi-agent model in which corporation agents and consumer agents interact with each other. We show that the mechanism to disseminate CSR is a positive feedback between the corporations popularity seeking behaviour and the consumer social learning in which CSR-seeking preference is evaluated according to both the local average of the preferences of surrounding consumers and the global average of the investment in CSR by all corporations. We also discuss an institutional design to establish CSR from an objectionable social state.

Keywords: CSR (corporate social responsibility), Quasi-public good, Institutional design, Positive Feedback, Multi-agent simulation

Joint work of: Hashimoto, Takashi; Shinohara, Naoto; Egashira, Susumu

Full Paper: http://drops.dagstuhl.de/opus/volltexte/2009/1911

Coherence-Driven Argumentation to Norm Consensus

Sindhu Joseph (IIIA - CSIC - Barcelona, ES)

In this paper coherence-based models are proposed as an alternative to logic-based BDI and argumentation models for the reasoning of normative agents. A model is provided for how two coherence-based agents can deliberate on how to regulate a domain of interest. First a deductive coherence model presented, in which the coherence values are derived from the deduction relation of an underlying logic; this makes it possible to identify the reasons for why a proposition is accepted or rejected. Then it is shown how coherence-driven agents can generate candidate norms for deliberation, after which a dialogue protocol for such deliberations is proposed. The resulting model is compared to current logic-based argumentation systems for deliberation over action.

Keywords: Deductive coherence, norm deliberation, normative agents, argumentation

Joint work of: Joseph, Sindhu; Prakken, Henry

Dynamics of acceptances in institutional contexts: a modal logic account

Emiliano Lorini (Université Paul Sabatier (IRIT) - Toulouse, FR)

We continue the work initiated in (Lorini et al. 2009; Lorini & Longin 2008), where the acceptance logic, a logic for modeling individual and collective acceptances was introduced. Here, we extend acceptance logic by two kinds of dynamic modal operators. The first kind consists of public announcements in institutional contexts. The second kind consists of acceptance shiftings: certain agents shift (change) their acceptances in order to accept a certain proposition qua members of a given institution. We show that the resulting logic has a complete axiomatisation in terms of reduction axioms for both dynamic operators

Keywords: Acceptance, institutions

Argumentation based Resolution of Conflicts Between Desires and Normative Goals

Sanjay Modgil (King's College - London, GB)

Norms represent what ought to be done, and their fulfillment can be seen as benefiting the overall system, society or organisation. However, individual agent goals (desire) may conflict with system norms. If a decision to comply with a norm is determined exclusively by an agent or, conversely, if norms are rigidly enforced, then system performance may be degraded, and individual agent goals may be inappropriately obstructed. To prevent such deleterious effects we propose a general framework for argumentation-based resolution of conflicts amongst desires and norms. In this framework, arguments for and against compliance are arguments justifying rewards, respectively punishments, exacted by 'enforcing' agents. The arguments are evaluated in a recent extension to Dung's abstract argumentation framework, in order that the agents can engage in metalevel argumentation as to whether the rewards and punishments have the required motivational force. We provide an example instantiation of the framework based on a logic programming formalism.

Keywords: Argumentation, Norms, Desires, Conflicts

Joint work of: Modgil, Sanjay; Luck, Michael

Full Paper: http://drops.dagstuhl.de/opus/volltexte/2009/1912

Partially Observable Markov Decision Processes with Behavioral Norms

Matthias Nickles (University of Bath, GB)

This extended abstract discusses various approaches to the constraining of Partially Observable Markov Decision Processes (POMDPs) using social norms and logical assertions in a dynamic logic framework. Whereas the exploitation of synergies among formal logic on the one hand and stochastic approaches and machine learning on the other is gaining significantly increasing interest since several years, most of the respective approaches fall into the category of relational learning in the widest sense, including inductive (stochastic) logic programming. In contrast, the use of formal knowledge (including knowledge about social norms) for the provision of hard constraints and prior knowledge for some stochastic learning or modeling task is much less frequently approached. Although we do not propose directly implementable technical solutions, it is hoped that this work is a useful contribution to a discussion about the usefulness and feasibility of approaches from norm research and formal logic in the context of stochastic behavioral models, and vice versa.

Keywords: Norms, Partially Observable Markov Decision Processes, Deontic Logic, Propositional Dynamic Logic

Joint work of: Nickles, Matthias; Rettinger, Achim

Full Paper: http://drops.dagstuhl.de/opus/volltexte/2009/1913

An essay on msic-systems

Jan Odelstad (University of Gävle, SE)

A theory of many-sorted implicative conceptual systems (abbreviated msic-systems) is outlined. Examples of msic-systems include legal systems, normative systems, systems of rules and instructions, and systems expressing policies and various kinds of scientific theories. In computer science, msic-systems can be used in, for instance, legal information systems, decision support systems, and multi-agent systems. In this essay, msic-systems are approached from a logical and algebraic perspective aiming at clarifying their structure and developing effective methods for representing them. Of special interest are the most narrow links or joinings between different strata in a system, that is between subsystems of different sorts of concepts, and the intermediate concepts intervening between such strata. Special emphasis is put on normative systems, and the role that intermediate concepts play in such systems, with an eye on knowledge representation issues. In this essay, normative concepts are constructed out of descriptive concepts using operators based on the Kanger-Lindahl theory of normative positions. An abstract architecture for a norm-regulated multi-agent system is suggested, containing a scheme for how normative positions will restrict the set of actions that the agents are permitted to choose from.

Keywords: Concept formation, Intermediary, Intermediate concept, Legal concept, Normative system, Normative position, Norm-regulated system, Agent architecture

Full Paper: http://drops.dagstuhl.de/opus/volltexte/2009/1914

Distrust is not Always the Complement of Trust (Position Paper)

Celia Costa Pereira (University of Milan, IT)

We believe that distrust can be as important as trust when agents are making a decision. An agent may not trust a source because of lack of positive evidence, but this does not necessarily mean the agent distrusts the source. Trust and distrust have to be considered as two separate concepts which can coexist.

We are aware that an adequate way to take this fact into account is by considering explicitly not only the agent's degree of trust in a source but also its independent degree of distrust. Explicitly taking distrust into account allows us to mark a clear difference between the distinct notions of negative trust and insufficient trust. More precisely, it is possible, unlike in approaches where only trust is explicitly accounted for, to "weigh" differently information from helpful, malicious, unknown, or neutral sources.

Keywords: Trust, Distrust, Decision Making

Full Paper: http://drops.dagstuhl.de/opus/volltexte/2009/1915

Early requirements engineering for e-customs decision support: Assessing overlap in mental models

Yao-Hua Tan (VU University Amsterdam, NL)

Developing decision support systems is a complex process. It involves stakeholders with diverging interpretations of the task and domain. In this paper, we propose to use ontology mapping to make a detailed analysis of the overlaps and differences between mental models of stakeholders. The technique is applied to an extensive case study about EU customs regulations. Companies which can demonstrate to be 'in control' of the safety and security in the supply chain, may become 'Authorized Economic Operator' (AEO), and avoid inspections by customs. We focus on a decision support tool, AEO Digiscan, developed to assist companies with an AEO self-assessment. We compared the mental models of customs officials, with mental models of the developers of the tool. The results highlight important differences in the interpretation of the new regulations, which will lead to adaptations of the tool.

Keywords: E-government, shared mental models, decision support systems

Joint work of: Burgemeestre, Brigitte; Liu, Jianwei; Hulstijn, Joris; Tan, Yao-Hua

Full Paper: http://drops.dagstuhl.de/opus/volltexte/2009/1916

Reflection and Norms: Towards a Model for Dynamic Adaptation for MAS

Ingo Timm (Goethe-Universität Frankfurt am Main, DE)

The design of self-organizing systems and particular multiagent systems (MAS) is a non trivial task. On the one hand the particular system should show a dynamic behavior according to its environment, to gain a central advantage of distributed systems, on the other hand it has to act on behalf of its user and the final results have to possess acceptable quality. Especially the quality of the overall system's behavior can become a critical issue, if the subsystems have their own objectives they have to optimize. In this paper we present a methodology that can be integrated into MAS for adapting their behavior allowing local optimization while respecting an acceptable level of the system's global goals.

Keywords: Balancing autonomy, multiagent simulation, manufacturing

Joint work of: Timm, Ingo J.; Lattner, Andreas D.; Schumann, Rene

Full Paper: http://drops.dagstuhl.de/opus/volltexte/2009/1917

Modeling and Validating Norms

Viviane Torres da Silva (University of Rio de Janeiro, BR)

Norms describe the permissions, prohibitions and obligations of agents in multi-agent systems in order to regulate their behavior. In this paper we propose a normative modeling language that makes possible the modeling of norms motivating the modeling of such norms together with the non-normative part of the system. In addition, we also propose a mechanism to validate the norms at design time, i.e., to check if the norms respect the constraints defined by the language and also their possible conflicts.

Keywords: Norm, modeling, validation, conflict, metamodel

Joint work of: Torres da Silva, Viviane; Braga, Christiano

Full Paper: http://drops.dagstuhl.de/opus/volltexte/2009/1918

A convention or (tacit) agreement betwixt us

Luca Tummolini (ISTC - CNR - Rome, IT)

The aim of this paper is to show that conventions are sources of tacit agreements. Such agreements are tacit in the sense that they are implicated by what the agents do (or forbear to do) though without that any communication between them be necessary. Conventions are sources of tacit agreements under two substantial assumptions: (1) that there is a salient interpretation, in some contexts, of every-one's silence as confirmatory of the others' expectations, and (2) that the agents share a value of not hostility. To characterize the normativity of agreements the Principle of Reliability is introduced.

Keywords: Agreement, convention, norm, pragmatics

Joint work of: Andrighetto, Giulia; Tummolini, Luca; Castelfranchi, Cristiano; Conte, Rosaria

Full Paper: http://drops.dagstuhl.de/opus/volltexte/2009/1919

Contract Formation through Preemptive Normative Conflict Resolution

Wamberto Vasconcelos (University of Aberdeen, GB)

We explore a rule-based formalisation for contracts: the rules capture conditional norms, that is, they describe situations arising during the enactment of a multi-agent system, and norms that arise from these situations. However, such rules may establish conflicting norms, that is, norms which simultaneously prohibit and oblige (or prohibit and permit) agents to perform particular actions. We propose to use a mechanism to detect and resolve normative conflicts in a preemptive fashion: these mechanisms are used to analyse a contract and suggest "amendments" to the clauses of the contract. These amendments narrow down the scope of influence of norms and avoid normative conflicts. Agents propose rules and their amendments, leading to a contract in which no conflicts may arise.

Keywords: Normative Conflict, Contracts

Joint work of: Vasconcelos, Wamberto; Norman, Timothy J.

Full Paper: http://drops.dagstuhl.de/opus/volltexte/2009/1920

Massively multiple online role playing games as normative multiagent systems

Harko Verhagen (Stockholm University, SE)

The latest advancements in computer games offer a domain of human and artificial agent behaviour well suited for analysis and development based on normative multi agent systems research. One of the most influential gaming trends today, Massively Multi Online Role Playing Games (MMORPG), poses new questions about the interaction between the players in the game. If we model the players and groups of players in these games as multiagent systems with the possibility to create norms and sanction norm violations we have to create a way to describe the different kind of norms that may appear in these situations. Certain situations in MMORPG are subject to discussions about how norms are created and propagated in a group, one such example involves the sleeper in the game Everquest, from Sony Online Entertainment (SOE). The Sleeper was at first designed to be unkillable, but after some events and some considerations from SOE the sleeper was finally killed. The most interesting aspect of the story about the sleeper is how we can interpret the norms being created in this example. We propose a framework to analyse the norms involved in the interaction between players and groups in MMORPG. We argue that our model adds complexity where we find earlier norm typologies lacking some descriptive power of this phenomenon, and we can even describe and understand the confusing event with the sleeper in Everquest.

Keywords: Norms, MMORPG

Joint work of: Magnus, Johansson; Verhagen, Harko

Full Paper: http://drops.dagstuhl.de/opus/volltexte/2009/1895

NorMAS-RE: a Normative Multiagent Approach to Requirements Engineering

Serena Villata (University of Torino, IT)

In this paper we present a new model, called NorMAS-RE, for the requirements analysis of a system. NorMAS-RE is a new model based on the multiagent systems paradigm with the aim to support the requirements analysis phase of systems design. This model offers a structured approach to requirements analysis, based on conceptual models defined following a visual modeling language, called dependence networks. The main elements of this visual language are the agents with their goals, capabilities and facts, similarly to the TROPOS methodology [10]. The normative component is present both in the ontology and in the conceptual metamodel, associating agents to roles they play inside the systems and a set of goals, capabilities and facts proper of these roles. This improvement

allows to define different types of dependence networks, called dynamic dependence networks and conditional dependence networks, representing the different phases of the requirements analysis of the system. This paper presents a requirements analysis model based on normative concepts such as obligation and institution.

The NorMAS-RE model is a model of semiformal specification featured by an ontology, a meta-model, a graphical notation and a set of constraints. Our model, moreover, allows the definition of the notion of coalition for the different kinds of network. We present our model using the scenario of virtual organizations based on a Grid network.

Keywords: Conditional dependence networks, obligations, sanctions, contrary to duty, requirements analysis

Full Paper: http://drops.dagstuhl.de/opus/volltexte/2009/1896

Normative Systems in Computer Science
Ten Guidelines for Normative Multiagent Systems

Guido Boella[1,2], Gabriella Pigozzi[2], and Leendert van der Torre[2]

[1] Department of Computer Science, University of Torino
[2] Computer Science and Communication, University of Luxembourg

Abstract. In this paper we introduce and discuss ten guidelines for the use of normative systems in computer science. We adopt a multiagent systems perspective, because norms are used to coordinate, organize, guide, regulate or control interaction among distributed autonomous systems. The first six guidelines are derived from the computer science literature. From the so-called 'normchange' definition of the first workshop on normative multiagent systems in 2005 we derive the guidelines to motivate which definition of normative multiagent system is used, to make explicit why norms are a kind of (soft) constraints deserving special analysis, and to explain why and how norms can be changed at runtime. From the so-called 'mechanism design' definition of the second workshop on normative multiagent systems in 2007 we derive the guidelines to discuss the use and role of norms as a mechanism in a game-theoretic setting, clarify the role of norms in the multiagent system, and to relate the notion of "norm" to the legal, social, or moral literature. The remaining four guidelines follow from the philosophical literature: use norms also to resolve dilemmas, and in general to coordinate, organize, guide, regulate or control interaction among agents, distinguish norms from obligations, prohibitions and permissions, use the deontic paradoxes only to illustrate the normative multiagent system, and consider regulative norms in relation to other kinds of norms and other social-cognitive computer science concepts.

1 Introduction

Normative systems are "systems in the behavior of which norms play a role and which need normative concepts in order to be described or specified" [36, preface]. There is an increasing interest in normative systems in the computer science community, due to the observation five years ago in the so-called AgentLink Roadmap [33, Fig. 7.1], a consensus document on the future of multiagent systems research, that norms must be introduced in agent technology in the medium term (i.e., now!) for infrastructure for open communities, reasoning in open environments and trust and reputation. However, there is no consensus yet in the emerging research area of normative multiagent systems on the kind of norms to be used, or the way to use them. Consider the following lines taken from a paper review report. A norm like "You should empty your plate" may be criticized, because it is not a (generic) norm but an obligation, or a sentence not presented

Dagstuhl Seminar Proceedings 09121
Normative Multi-Agent Systems
http://drops.dagstuhl.de/opus/volltexte/2009/1902

as a norm, such as an imperative or command like “Empty your plate!”, may be criticized because it is a norm. Alternatively, a proposed normative multiagent systems may be criticized by a reviewer, because, for example, norms cannot be violated, norms cannot be changed, and so on. These criticisms suggest that more agreement on the use of norms and normative systems in computer science would be useful.

The research question of this paper is to give general guidelines for the use of “norms” and “normative systems” in computer science. During the past two decades normative systems have been studied in a research field called deontic logic in computer science (ΔEON), and normative multiagent systems may be seen as the research field where the traditional normative systems and ΔEON meet agent research. In these areas, the following two related challenges emerged to a common use of “norms” and “normative systems” in computer science.

There are many distinct notions of “normative systems” in the literature due to the use of the concept “norm” in distinct disciplines, just like there are many definitions of “agent” or “actor” due to its use across disciplines. Traditionally normative systems have been studied in philosophy, sociology, law, and ethics, and “norms” can therefore be, for example, social expectations, legal laws or linguistic imperatives or commands.

The role of norms in computer science is changing and solutions based on multiagent systems are increasing. The seventh ΔEON conference [31, 32] in 2004 in Madeira, Portugal, had as special theme “deontic logic and multiagent systems,” the eighth ΔEON conference in 2006 in Utrecht, the Netherlands, had as special focus “artificial normative systems” [22, 21], and the ninth ΔEON conference [22, 43] in Luxembourg in 2008 was co-located with the third workshop on normative multiagent systems NorMAS. Gradually the ΔEON research focus changes from logical relations among norms to, for example, agent decision making, and to systems in which norms are created and in which agents can play the role of legislators.

We approach this question of defining guidelines for normative multiagent system research by first considering two consensus definitions in the computer science literature of previous normative multiagent systems NorMAS workshops, from which we derive our first six guidelines. The remaining four guidelines follow from a short survey of the philosophical literature.

2 Normative multiagent systems

Before we consider the ‘normchange’ and ‘mechanism design’ definition of normative multiagent systems, we start with a dictionary definition of normative systems. With ‘normative’ we mean ‘conforming to or based on norms’, as in *normative behavior* or *normative judgments*. According to the Merriam-Webster Online [35] Dictionary, other meanings of normative not considered here are ‘of, relating to, or determining norms or standards’, as in *normative tests*, or ‘prescribing norms’, as in *normative rules of ethics* or *normative grammar*. With ‘norm’ we mean ‘a principle of right action binding upon the members of a group and serving to guide, control, or regulate proper and acceptable behavior’. Other meanings of ‘norm’ given by the Merriam-Webster Online Dictionary but not considered here are ‘an authoritative standard or model’, ‘an average like a standard, typical pattern, widespread practice or rule in a group’, and various definitions used in mathematics.

2.1 The normchange definition

The first definition of a normative multiagent system emerged after two days of discussion at the first workshop on normative multiagent systems NorMAS held in 2005 as a symposium of the Artificial Intelligence and Simulation of Behaviour convention (AISB) in Hatfield, United Kingdom:

The normchange definition. “A normative multiagent system is a multiagent system together with normative systems in which agents on the one hand can decide whether to follow the explicitly represented norms, and on the other the normative systems specify how and in which extent the agents can modify the norms” [8].

The first three guidelines are derived from this definition. The first one concerns the explicit representation of norms, which has been interpreted either that norms must be explicitly represented in the system (the ‘strong’ interpretation) or that norms must be explicitly represented in the system specification (the ‘weak’ interpretation). The first guideline is to make explicit and motivate which interpretation is used, the strong one, the weak one, or none of them.

Guideline 1 *Motivate which definition of normative multiagent system is used.*

The motivation for the strong interpretation of the explicit representation is to prevent a too general notion of norms. Any requirement can be seen as a norm the system has to comply with; but why should we do so? Calling every requirement a norm makes the concept empty and useless. The weak interpretation is used to study the following two important problems in normative multiagent systems.

Norm compliance. How to decide whether systems or organizations comply with relevant laws and regulations? For example, is a hospital organized according to medical regulations? Does a bank comply with Basel 2 regulations?

Norm implementation. How can we design a system such that it complies with a given set of norms? For example, how to design an auction such that agents cannot cooperate?

The second guideline follows from the fact that agents can decide whether to follow the norms. This part of the definition is borrowed from the ΔEON tradition, whose founding fathers Meyer and Wieringa observe that “until recently in specifications of systems in computational environments the distinction between normative behavior (as it *should be*) and actual behavior (as it *is*) has been disregarded: mostly it is not possible to specify that some system behavior is non-normative (illegal) but nevertheless possible. Often illegal behavior is just ruled out by specification, although it is very important to be able to specify what should happen if such illegal but possible behaviors occurs!" [36, preface]. However, constraints are well studied and well understood concepts, so if a norm is a kind of constraint, the question immediately is raised what is special about them.

Guideline 2 *Make explicit why your norms are a kind of (soft) constraints that deserve special analysis.*

Examples of issues which have been analyzed for norms but to a less degree for other kinds of constraints are ways to deal with violations, representation of permissive norms, the evolution of norms over time (in deontic logic), the relation between the cognitive abilities of agents and the global properties of norms, how agents can acquire norms, how agents can violate norms, how an agent can be autonomous [17] (in normative agent architectures and decision making), how norms are created by a legislator, emerge spontaneously or are negotiated among the agents, how norms are enforced, how constitutive or counts-as norms are used to describe institutions, how norms are related to other social and legal concepts, how norms structure organizations, how norms coordinate groups and societies, how contracts are related to contract frames and contract law, how legal courts are related, and how normative systems interact?

For example, the norms of global policies may be represented as soft constraints, which are used in detective control systems where violations can be detected, instead of hard constraints restricted to preventative control systems in which violations are impossible. The typical example of the former is that you can enter a train without a ticket, but you may be checked and sanctioned, and an example of the latter is that you cannot enter a metro station without a ticket. However, if the norms are represented as constraints, then how to analyze that detective control is the result of actions of agents and therefore subject to errors and influenceable by actions of other agents? For example, it may be the case that violations are not often enough detected, that law enforcement is lazy or can be bribed, there are conflicting obligations in the normative system, that agents are able to block the sanction, block the prosecution, update the normative system, etc.

The third guideline follows from the fact that norms can be changed by the agents or the system, which distinguished this definition of normative multiagent system from the common framework used in the ΔEON community, and led to the identification of this definition as the "normchange" definition of normative multiagent systems.

Guideline 3 *Explain why and how norms can be changed at runtime.*

For example, a norm can be made by an agent, as legislators do in a legal system, or there can be an algorithm that observes agent behavior, and suggests a norm when it observes a pattern. The agents can vote on the acceptance of the norm. Likewise, if the system observes that a norm is often violated, then apparently the norm does not work as desired, and it undermines the trust of the agents in the normative system, so the system can suggest that the agents can vote whether to retract or change the norm.

2.2 The mechanism design definition

The fourth, fifth and sixth guideline follow from the consensus definition of the second workshop on normative multiagent systems NorMAS held as Dagstuhl Seminar 07122 in 2007. After four days of discussion, the participants agreed to the following consensus definition:

The mechanism design definition. "A normative multiagent system is a multiagent system organized by means of mechanisms to represent, communicate, distribute, detect, create, modify, and enforce norms, and mechanisms to deliberate about norms and detect norm violation and fulfilment." [10]

The fourth guideline emphasizes the game-theoretic model and the notion of a norm as a mechanism. According to Boella *et al.*, "the emphasis has shifted from representation issues to the mechanisms used by agents to coordinate themselves, and in general to organize the multiagent system. Norms are communicated, for example, since agents in open systems can join a multiagent system whose norms are not known. Norms are distributed among agents, for example, since when new norms emerge the agent could find a new coalition to achieve its goals. Norm violations and norm compliance are detected, for example, since spontaneous emergence norms of among agents implies that norm enforcement cannot be delegated to the multiagent infrastructure." [10]

Guideline 4 *Discuss the use and role of norms always as a mechanism in a game-theoretic setting.*

Here we refer to game theory in a very liberal sense, not only to classical game theory studied in economics, which has been criticized for its ideality assumptions. Of particular interest are alternatives taking the limited or bounded rationality of decision makers into account. For example, Newell [37] and others develop theories in artificial intelligence and agent theory, replace probabilities and utilities by informational (knowledge, belief) and motivational attitudes (goal, desire), and the decision rule by a process of deliberation. Bratman [11] further extends such theories with intentions for sequential decisions and norms for multiagent decision making. Alternatively, Gmytrasiewitcz and Durfee [19] replace the equilibria analysis in game theory by recursive modelling, which considers the practical limitations of agents in realistic settings such as acquiring knowledge and reasoning so that an agent can build only a finite nesting of models about other agents' decisions.

Games can explain that norms should satisfy various properties to be effective as a mechanism to obtain desirable behavior. For example, the system should not sanction without reason, as for example Caligula or Nero did in the ancient Roman times, as the norms would loose their force to motivate agents. Moreover, sanctions should not be too low, but they also should not be too high, as shown by argument of Beccaria. Otherwise, once a norm is violated, there is no way to prevent further norm violations.

Games can explain also the role of various kinds of norms in a system. For example, assume that norms are added to the system one after the other and this operation is performed by different authorities at different levels of the hierarchy. Lewis "master and slave" game [30] shows that the notion of permission alone is not enough to build a normative system, because only obligations divide the possible actions into two categories or spheres: the sphere of prohibited actions and the sphere of permitted (i.e., not forbidden) actions or "the sphere of permissibility". More importantly, Bulygin [13] explains why permissive norms are needed in normative systems using his "Rex, Minister and Subject" game. "Suppose that Rex, tired of governing alone, decides one day to appoint a Minister and to endow him with legislative power. [...] an action commanded by Minister becomes as obligatory as if it would have been commanded by Rex. But Minister has no competence to alter the commands and permissions given by Rex." If Rex permits hunting on Saturday and then Minister prohibits it for the whole week, its prohibition on Saturday remains with no effect.

As another example, in our game theoretic approach to normative systems [9] we study the following kind of normative games.

Violation games: interacting with normative systems, obligation mechanism, with applications in trust, fraud and deception.

Institutionalized games: counts-as mechanism, with applications in distributed systems, grid, p2p, virtual communities.

Negotiation games: MAS interaction in a normative system, norm creation action mechanism, with applications in electronic commerce and contracting.

Norm creation games: multiagent system structure of a normative system, permission mechanism, with applications in legal theory.

Control games: interaction among normative systems, nested norms mechanism, with applications in security and secure knowledge management systems.

The fifth guideline follows from the introduction of organizational issues in the definition of normative multiagent systems. Norms are no longer seen as the mechanism to regulate behavior of the system, but part of a larger institution. This raises the question what precisely the role of norms is in such an organization.

Guideline 5 *Clarify the role of norms in your system.*

Norms are rules used to guide, control, or regulate desired system behavior. However, this is not unproblematic. For example, consider solving traffic problems by introducing norms, as a cheap alternative to building new roads. It does not work, for the following two reasons. The first reason is that if you change the system by building new norms or introducing new norms, then people will adjust their behavior. For example, when roads improve, people tend to live further away from their work. In other words, a normative multiagent system is a self-organizing system. Moreover, the second problem with norm design is that norms can be violated. For example, most traffic is short distance, for which we could forbid using the car. However, it is hard to enforce such a norm, since people will always claim to have come from long distance, even if they live around the corner.

Norms can also be seen as one of the possible incentives to motivate agents, which brings us again back to economics.

> "Economics is, at root, the study of incentives: how people get what they want, or need, especially when other people want or need the same thing. Economists love incentives. They love to dream them up and enact them, study them and tinker with them. The typical economist believes the world has not yet invented a problem that he cannot fix if given a free hand to design the proper incentive scheme. His solution may not always be pretty–but the original problem, rest assured, will be fixed. An incentive is a bullet, a lever, a key: an often tiny object with astonishing power to change a situation.
> ...
> There are three basic flavors of incentive: economic, social, and moral. Very often a single incentive scheme will include all three varieties. Think about the anti-smoking campaign of recent years. The addition of $3-per-pack "sin tax" is a strong economic incentive against buying cigarettes. The banning of cigarettes in restaurants and bars is a powerful social incentive. And when the U.S. government asserts that terrorists raise money by selling black-market cigarettes, that acts as a rather jarring moral incentive.' [29]

Here it is important to see that moral incentives are very different from financial incentives. For example, Levitt [29, p.18-20], discussing an example of Gneezy and Rustichini [20], explains that the number of violations may *increase* when financial sanctions are imposed, because the moral incentive to comply with the norm is destroyed. The fact that norms can be used as a mechanism to obtain desirable system behavior, i.e. that norms can be used as incentives for agents, implies that in some circumstances economic incentives are not sufficient to obtain such behavior. For example, in a widely discussed example of the so-called centipede game, there is a pile of thousand pennies, and two agents can in turn either take one or two pennies. If an agent takes one then the other agent takes turn, if it takes two then the game ends. A backward induction argument implies that it is rational only to take two at the first turn. Norms and trust have been discussed to analyze this behavior, see [28] for a discussion.

A rather different role of norms is to organize systems. To manage properly complex systems like multiagent systems, it is necessary that they have a modular design. While in traditional software systems, modularity is addressed via the notions of class and object, in multiagent systems the notion of organization is borrowed from the ontology of social systems. Organizing a multiagent system allows to decompose it and defining different levels of abstraction when designing it. Norms are another answer to the question of how to model organizations as first class citizens in multiagent systems. Norms are not usually addressed to individual agents, but rather they are addressed to roles played by agents [6]. In this way, norms from a mechanism to obtain the behavior of agents, also become a mechanism to create the organizational structure of multiagent systems. The aim of an organizational structure is to coordinate the behavior of agents so to perform complex tasks which cannot be done by individual agents. In organizing a system all types of norms are necessary, in particular, constitutive norms, which are used to assign powers to agents playing roles inside the organization. Such powers allow to give commands to other agents, make formal communications and to restructure the organization itself, for example, by managing the assignment of agents to roles. Moreover, normative systems allow to model also the structure of an organization and not only the interdependencies among the agents of an organization. Consider a simple example from organizational theory in Economics: an enterprise which is composed by a direction area and a production area. The direction area is composed by the CEO and the board. The board is composed by a set of administrators. The production area is composed by two production units; each production unit by a set of workers. The direction area, the board, the production area and the production units are functional areas. In particular, the direction area and the production areas belong to the organization, the board to the direction area, etc. The CEO, the administrators and the members of the production units are roles, each one belonging to a functional area, e.g., the CEO is part of the direction area. This recursive decomposition terminates with roles: roles, unlike organizations and functional areas, are not composed by further social entities. Rather, roles are played by other agents, real agents (human or software) who have to act as expected by their role. Each of these elements can be seen as an institution in a normative system, where legal institutions are defined by Ruiter [39] as "systems of [regulative and constitutive] rules that provide frameworks for social action within larger rule-governed settings". They are "relatively independent institutional legal orders within the comprehensive legal orders".

The sixth guideline follows from the trend towards a more dynamic interactionist view identified at the second NorMAS workshop. "This shift of interest marks the passage of focus from the more static legalistic view of norms (where power structures are fixed) to the more dynamic interactionist view of norms (where agent interaction is the base for norm related regulation)." This ties in to what Strauss [42] called "negotiated order", Goffman's [23] view on institutions, and Giddens' [18] structuration theory. The two views are summarized in Table 1. For example, if in a normative system the norms are created by agents it is more a legalistic view, but if there is an algorithm that observes behavior and proposes norms, it is more an interactionist view. The latter procedure can still be put up to vote for the agents, and being accepted or rejected. As another example, suppose a monitoring system observes that some norms are violated frequently, then it can propose to delete the norms, for example because the violations decrease the trust of the agents in the system.

	Legalistic view	Interactionist view
	top-down view	bottom-up view
normative system	regulatory instrument to regulate emerging behavior of open systems	autonomous individually oriented view regularities of behavior emerge without any enforcement system
compliance	sanctions	sharing of the norms their goals happen to coincide they feel themselves as part of the group they share the same values sanctions are not always necessary social blame and spontaneous exclusion
freedom to create norms	restricted to contracts	emergence of norms

Table 1. Two views on normative multiagent systems

Guideline 6 *Relate the notion of "norm" to the legal, social, or moral literature.*

Boella *et al.* put the legalistic and interactionist view in the context of five levels in the development of normative multiagent systems, summarized in Table 2. They observe that "for each level the development of the normative multiagent system will take a much larger effort than the development of similar systems at lower levels." For example, if norms are explicitly represented (level 2) rather than built into the system (level 1), then the system has to be much more flexible to deal with the variety of normative systems that may emerge. However, it may be expected that normative multiagent systems realized at higher levels will have a huge effect on social interaction, in particular on the web" [10]. We illustrate the more dynamic interactionist viewpoint on normative multiagent systems using virtual communities in virtual reality settings like Second Life. In these virtual communities, human agents interact with artificial agents in a virtual world. This interactionist view, which has been promoted in the multiagent systems community by Cristiano Castelfranchi [14], becomes essential in applications related to virtual communities. In Second Life, for example, communities emerge in which the behavior of its members show increasing homogeneity.

level		
1	off-line norm design [41]	norms are imposed by the designer and automatically enforced, agents cannot organize themselves by means of norms
2	norm representation	norms are explicitly represented they can be used in agent communication and negotiation a simple kind of organizations and institutions can be created
3	norm manipulation	a legal reality is created agents can add and remove norms following the rules of the normative system
4	social reality	the ten challenges discussed Table 3
5	moral reality	This goes beyond present studies in machine ethics [4]

Table 2. Five levels in the development of normative multiagent systems. [10]

Boella *et al.* also mention ten challenges posed by the interactionist viewpoint: They “take the perspective from an agent programmer, and consider which kinds of tools like programming primitives, infrastructures, protocols, and mechanisms she needs to deal with norms in the example scenario. Similar needs exist at the requirements analysis level, or the design level, but we have chosen for the programming level since it makes the discussion more concrete, and this level is often ignored when norms are discussed. The list is not exhaustive, and there is some overlap between the challenges. Our aim is to illustrate the range of topics which have to be studied, and we therefore do not attempt to be complete” [10].

Challenge	Tool
1	Tools for agents supporting communities in their task of recognizing, creating, and communicating norms to agents
2	Tools for agents to simplify normative systems, recognize when norms have become redundant, and to remove norms
3	Tools for agents to enforce norms
4	Tools for agents to preserve their autonomy
5	Tools for agents to construct organizations
6	Tools for agents to create intermediate concepts and normative ontology, for example to decide about normative gaps
7	Tools for agents to decide about norm conflicts
8	Tools for agents to voluntarily give up some norm autonomy by allowing automated norm processing in agent acting and decision making
9	Tools for conviviality
10	Tools for legal responsibility of the agents and their principals

Table 3. Ten challenges posed by the interactionist viewpoint. [10]

3 Philosophical foundations

We consider only four guidelines from the rich history of deontic logic in philosophical logic. The first two guidelines follow from the history of deontic logic, the third guideline from the methodology in deontic logic based on deontic paradoxes, and the fourth guideline from the deontic logic in computer science to study norms in the way they interact with other concepts. We believe philosophical logic has much more to offer for computer scientists, but we restrict ourselves to the most important issues.

3.1 Deontic logic

In 1951, the philosopher and logician Von Wright wrote a paper called “deontic logic” [45], which subsequently became the name of the research area concerned with normative concepts such as obligation, prohibition and permission. The term deontic is derived from the ancient Greek déon, meaning that which is binding or proper. The basis of his formal system was an observed relation between obligation and permission. For example, he defined the obligation to tell the truth by interpreting that it is good to tell the truth, and therefore it is bad to lie. If it is bad to lie then it is forbidden to lie, and therefore it is not permitted to lie. Summarizing, something is obligatory when its absence is not permitted. This logical relation is based on the binary distinction between good and bad, as illustrated by its possible worlds semantics distinguishing between good and bad worlds.

The relation between obligation and violation was given by Anderson seven years later in 1958, in a paper called “A Reduction of Deontic Logic to Alethic Modal Logic” [3]. In this paper, he proposed a reduction of obligation to violation. For example, the obligation to tell the truth means that a lie necessarily implies a violation. In general, and in its simplest form, something is obliged if and only if its absence necessarily leads to a violation.

The problems of these early approaches were illustrated in 1963 in a paper by Chisholm called “Contrary-to-duty imperatives and deontic logic” [16]. Consider a pregnant woman going to the hospital. The shortest way to go to the hospital is turning left, which obviously is what the driver is doing. However, there is a norm that it is forbidden to go to the left, so she is violating the obligation to go to the right. Now, the problem is due to two additional norms. One says that if she goes to the left she has to signal that she is going to the left, and one says that if she goes to the right she has to signal that she is going to the right. The problem here is how to explain that given that she is going to the left, she is obliged to signal that she is going to the left. This obligation cannot be explained by the basic distinction between good and bad, because the good thing here is to go to the right and signaling that she is going to the right at least from the perspective of traffic law.

Modern deontic logic started with a paper by Bengt Hansson in 1969, called “An Analysis of some Deontic Logics” [27]. In this paper he introduced a semantics based on a betterness relation for conditional obligations (it was axiomatized only six years later). With his paper he started modern deontic logic. The pregnant woman example can be represented by an ideal situation from the perspective of traffic law – in which the car goes to the right and signals that it will go to the right, but the situation in which the car goes to the left and signals that it will go to the left is better to the one in which the car goes to the left but signals that it will go to the right. As mentioned in the previous section, it is precisely the possibility of violation, that led Meyer and Wieringa to introduce the use of norms and deontic logic in computer science. Moreover, the formalism has become popular also in other areas of computer science too, such as non-monotonic logic and qualitative decision theory.

The seventh guideline says not to use the prehistory of deontic logic in the fifties and sixties of the previous century, but adapt to modern deontic logic as studied since the seventies. To say it crudely, Von Wright’s and Anderson’s systems have not been in the philosophical literature for forty years, so there seems little reason for computer scientists to return to these forgotten theories.

Guideline 7 *Use norms not only to distinguish right from wrong, but also to resolve dilemmas, and use norms not only describe violations, but in general to coordinate, organize, guide, regulate or control interaction among agents.*

Von Wright's system became known as the 'Old System', since he developed many modern systems too. Whereas the old system was based on monadic modal logic, the new systems were based on dyadic modal logics, just like Hansson's peference-based deontic logic. However, some people started to call the old system 'Standard Deontic Logic' or SDL, and this led to a lot of confusion. Some people in computer science, maybe due to the important role of standards in this research field, believed that a system called Standard Deontic Logic has to be a common reference for future explorations. To emphasize this misconception, let us consider some more examples. Remember that SDL sees norms just as being good and bad, or right and wrong. For example, it is right to obey your parents, it is wrong to hijack a plane, it is good to finish in time, and it is bad to write a computer virus. Though this is one way to look at norms, it is often not sufficient. Consider the following example. Suppose there is a plane hijacked by terrorists heading towards some high towers. There is a moral dilemma whether we may or should shoot down this plane. It is a dilemma, because if we shoot down the plane there will be a lot of innocent people killed, but if we don't shoot down the plane, then the plane will crash into these buildings. So it is a moral dilemma, and just thinking about right and wrong is not sufficient to solve it. People have been thinking about this kind of problems in ethics, and there are different theories. For example, a utilitarian theory says that you should minimize the damage. So what you should do is shoot down this plane, you may do it, you are obliged to do it, because if you don't do it, then the number of casualties will be higher than if you don't shoot it down. However, another ethical theory says that we may not shoot down the plane, because it is active involvement of ourselves, and if we do this, then we are responsible for killing the people in the plane. So it is forbidden to shoot down the plane. If we represent norms in computer systems, as we are now starting to do, then we can expect to find conflicts, and we therefore need to have a way to resolve these conflicts.

Anderson's reduction suggests that a norm is in the end just a description of violations. In the previous example of hijacking a plane, a norm says what counts as hijacking a plane, we call it a legal ontology, there is a norm telling us that it is a violation to hijack a plane, and there is a sanction associated when you do this. This is also very popular in computer science, but it is also insufficient. One indication of the problems related to this kind of reduction, is that people have not been able to give a reduction from Hansson's dyadic deontic logic to violations. Another conceptual problem is that violation is associated with norms and imperatives instead of obligations and prohibitions studied in deontic logic, an issue we discuss further below. But there are also practical problems. Consider for example the much discussed European Constitution. The question is, can we look at this text only as a set of descriptions of norm violations? There is an organizational structure, distribution of powers, there are several norms, there are several sanctions, and it seems, at least at first sight, that we can try to represent it as a set of norm violations. The problem with this constitution is that it has been rejected, we will never know what it really means. What we can do is look at the rules that are in force in the European Union, of which one says that the national deficit of a country should be below 3% of the national gross product. However, there were various countries who broke this norm, France and Germany in particular. However, the violation was not recognized, and the countries were not sanctioned.

The latter point is a little more subtle, because there are systems with violation predicates which are useful to reason about normative systems, namely diagnostic theories. Such theories have been developed in the eighties in artificial intelligence and computer science, and have been applied to a wide variety of domains, such as fault diagnosis of systems, or medical diagnosis. They can also be used to diagnose a court case, and determine whether someone is guilty or not (well, in principle, there are some more issues in legal reasoning which we will not consider here). However, as is well-known in ΔEON community, such a system is neither a deontic logic, nor a normative system. The main problem of such a formal system is that it does not deal easily with consequences of violations, so-called contrary-to-duty reasoning. For example, agent A should do α, and if he does not do so, then police agent B should punish him (a standard example in which norms regulate interaction between two agents). For a further discussion on this approach and its limitations, see [44].

The seventh guideline follows from the more recent literature on deontic logic, of which we have given a very sketchy overview in Table 4. In the beginning deontic logic was syntax based, and semantics came later. Modal semantics became very popular for some time, but during the past twenty years approaches based on non-monotonic logic and imperatives have become more and more popular. Nowadays, we can no longer say that deontic logic is a branch of modal logic. The use of possible worlds (Kripke) semantics is useful to distinguish good from bad, but less useful to represent dilemmas, or imperatives.

period	tradition	main issue
50s	monadic modal logic	relation O and P
60s	dyadic modal logic	relation O and facts, violations, sub-ideality and optimality, CTD
70s	temporal deontic logic	relation O and time
80s	action deontic logic	relation O and actions
90s	defeasible deontic logic	dilemmas, CTD
00s	imperatives, normative systems	Jorgensen's dilemma

Table 4. A schematic reconstruction of deontic logic

In particular, the seventh guideline follows from attempts during the past decade to base the semantics of deontic logic on imperatives. Deontic logic describes logical relations between obligations, prohibitions and permissions, but it is conditional on a normative system, which is typically left implicit. More precisely, there are two distinct philosophical traditions, the one of deontic logic discussed thus far, and another one of normative systems. The main challenge during the past ten years in deontic logic is how these two traditions can be merged. This story is explained in [25], which at this moment is the best introduction to current research in deontic logic. The most famous proponents of normative systems are Alchourrón and Bulygin [1], who argue in 1971 that a normative system should not be defined as a set of norms, as is commonly done, but in terms of consequences:

> "When a deductive correlation is such that the first sentence of the ordered pair is a case and the second is a solution, it will be called normative. If among the deductive correlations of the set α there is at least one normative correlation, we shall say that the set α has normative consequences. A system of sentences which has some normative consequences will be called a normative system." [1, p.55].

All the famous deontic logicians have discussed this subtle issue, often introducing new terminology. For example, Von Wright distinguished norms and normative propositions, and Alchourrón distinguished prescriptive and descriptive obligations.

Guideline 8 *Distinguish norms from obligations, prohibitions and permissions.*

As an example, consider the input/output logic framework introduced by Makinson and van der Torre [34]. The first input/output logic principle is that norms are not represented by propositional sentences, as in AGM framework for theory change [2], or as modal formulas, as in deontic logic, but as pairs of formulas of an arbitrary logic. The pair of propositional formulas represents a rule, and the two propositional formulas are called the antecedent and consequent of the rule. The second principle of the input/output logic framework is that the primary role of norms in a normative system is the derivation of obligations and prohibitions. Which obligations and prohibitions can be derived from a normative system depends on the factual situation, which we call the *context* or *input* and represent by a propositional formula. The function that associates with each context the set of obligations describes the meaning of the normative system, because it is a kind of 'operational semantics' of the normative system. An input/output operation $out : (2^{L \times L}) \times L \to 2^L$ is a function from the set of normative systems and contexts, to a set of sentences of L. We say that x is obligatory in normative system N and context a if $x \in out(N, a)$. The simplest input/output logic defined by Makinson and van der Torre is so-called simple-minded output. x is in the simple-minded output of N in context a, written as $x \in out_1(N, a)$, if there is a set of norms $(a_1, x_1), \ldots, (a_n, x_n) \in N$ such that $a_i \in Cn(a)$ and $x \in Cn(x_1 \wedge \ldots \wedge x_n)$, where $Cn(S)$ is the consequence set of S in L. Such an operational semantics can be axiomatized as follows. $out_1(N)$ is the minimal set that contains $N \cup \{(\top, \top)\}$, is closed under replacement of logical equivalents in antecedent and consequent, and the following proof rules strengthening of the input SI, weakening of the output WO, and conjunction rule AND.

$$\frac{(a, x)}{(a \wedge b, x)} SI \qquad \frac{(a, x \wedge y)}{(a, x)} WO \qquad \frac{(a, x), (a, y)}{(a, x \wedge y)} AND$$

Ten philosophical problems on deontic logic are given by Hansen *et al.* [26] and listed in Table 5. Of this list, we observe that constitutive norms and intermediate concepts are often seen as the same problem (though constitutive norms can be used also for other problems than intermediate concepts), and that there are other problems not listed in this paper, such as the equivalence of normative systems, or redundancy of a norm in a normative system [12, 5].

1. How can deontic logic be reconstructed in accord with the philosophical position that norms are neither true nor false?
2. When is a set of norms to be termed 'coherent'?
3. How can deontic logic accommodate possible conflicts of norms? How can the resolution of apparent conflicts be semantically modeled?
4. How do we reason with contrary-to-duty obligations which are in force only in case of norm violations?
5. How to define dyadic deontic operators with regard to given sets of norms and facts?
6. How to distinguish various kinds of permissions and relate them to obligations?
7. How can meaning postulates and intermediate terms be modeled in semantics for deontic logic reasoning?
8. How to define counts-as conditionals and relate them to obligations and permissions?
9. How to revise a set of regulations or obligations? Does belief revision offer a satisfactory framework for norm revision? Can the belief merging framework deal with the problem of merging sets of norms?

Table 5. Ten philosophical problems. "We argue that norms, not ideality, should take the central position in deontic semantics, and that a semantics that represents norms, as input/output logic does, provides helpful tools for analyzing, clarifying and solving the problems of deontic logic." [26]

3.2 Methodology

Not surprisingly for such a highly simplified theory like Von Wright's Old System, also know as SDL, there are many features of actual normative reasoning that SDL does not capture. Notorious are the so-called 'paradoxes of deontic logic', which are usually dismissed as consequences of the simplifications of SDL. For example, Ross's paradox [38], the counterintuitive derivation of "you ought to mail or burn the letter" from "you ought to mail the letter", is typically viewed as a side effect of the interpretation of 'or' in natural language.

Guideline 9 *Don't motivate your new theory by toy "paradoxical" examples, but use the deontic paradoxes to illustrate basic properties of your system.*

Computer scientists are usually surprised when they read the philosophical literature, because the posed problems seem to have a trivial solution. For example, the most famous deontic paradox of all, is often posed as the problem to give a consistent representation such that none of the sentences can be derived from the others:

1. A certain man should go to the assistance of his neighbors,
2. If he goes, he should tell them he is coming
3. If he does not go, he should not tell them that he is coming
4. He does not go.

In SDL the set $\{Oa, O(a \rightarrow t), \neg a \rightarrow O(\neg t), \neg a\}$ is inconsistent, and in $\{Oa, a \rightarrow O(t), \neg a \rightarrow O(\neg t), \neg a\}$ the sentences are not logically independent. However, this problem is trivially solved by replacing the material implication by a strict implication, or a relevant implication, or a defeasible implication. This has been known since the early days of deontic logic, as may be expected since both deontic logic and conditional logic were major branches of philosophical logic. In general, any paradoxical consequence can be solved by simply weakening the logic (e.g., solve Ross' paradox by replacing standard deontic logic by a non-normal modal logic). The misconception is simply due to the fact that the deontic paradoxes do not work the same way as experiments in engineering or the sciences. They are just used to illustrate the formal system, not to guide research in the area. A similar phenomena and misconception has been present in the field of defeasible reasoning and deontic logic, where the use of the infamous Tweety example has been criticized for similar reasons. The reason why contrary-to-duty paradoxes have been discussed for fifty years in deontic logic is that a lot of normative reasoning is directly or indirectly related to violations, just like in defeasible reasoning a lot of reasoning is directly or indirectly related to exceptions.

3.3 From philosophy to computer science

Most of the confusions in deontic logic are due to the abstract nature of the formal systems. In areas of computer science like multiagent systems or knowledge representation we need to be more detailed, and most of the problems then disappear.

Guideline 10 *Regulative norms should not be considered by themselves, but in relation to permissive norms, constitutive norms, procedural norms, agents, roles, groups, societies, rights, duties, obligations, time, beliefs, desires, intentions, goals, roles, and other kinds of norms and other social-cognitive computer science concepts.*

Regulative norms specify the ideal and varying degrees of sub-ideal behavior of a system by means of obligations, prohibitions and permissions. Constitutive norms are based on the notion that “X counts-as Y in context C” and are used to support regulative norms by introducing institutional facts in the representation of legal reality. The notion of counts-as introduced by Searle [40] has been interpreted in deontic logic in different ways and it seems to refer to different albeit related phenomena [24]. Substantive norms define the legal relationships of people with other people and the state in terms of regulative and constitutive norms, where regulative norms are obligations, prohibitions and permissions, and constitutive norms state what counts as institutional facts in a normative system. Procedural norms are instrumental norms, addressed to agents playing roles in the normative system, which aim at achieving the social order specified in terms of substantive norms [7].

4 Summary

Next generation normative multiagent systems contain general and domain independent norms by combining three existing representations of normative multiagent systems. First, theories of normative systems and deontic logic, the logic of obligations and permissions, for the explicit representation of norms as rules, the application of such rules, contrary-to-duty reasoning and the relation to permissions. Second, agent architecture for software engineering of agents and a model of normative decision making. Third, a game-theoretic approach for model of interaction explaining the relation among social norms and obligations, relating regulative norms to constitutive norms, the evolution of normative systems, and much more. In this paper, we introduce and discuss ten guidelines for the development of normative multiagent systems.

1. Motivate which definition of normative multiagent system is used.
2. Make explicit why norms are a kind of (soft) constraints deserving special analysis.
3. Explain why and how norms can be changed at runtime.
4. Discuss the use and role of norms as a mechanism in a game-theoretic setting.
5. Clarify the role of norms in the multiagent system.
6. Relate the notion of “norm” to the legal, social, or moral literature.
7. Use norms not only to distinguish right from wrong, but also to resolve dilemmas, and use norms not only describe violations, but in general to coordinate, organize, guide, regulate or control interaction among agents.
8. Distinguish norms from obligations, prohibitions and permissions.
9. Use the deontic paradoxes only to illustrate the normative multiagent system.
10. Consider regulative norms in relation to other kinds of norms and concepts.

Table 6. Ten guidelines for the development of normative multiagent systems

The use of norms and normative systems in computer science are examples of the use of social concepts in computer science, which is now so well-established that the original meaning of some of these concepts in the social sciences is sometimes forgotten. For example, the original meaning of a “service” in business economics is rarely considered by computer scientists working on service oriented architectures or web services, and likewise for service level agreements and contracts, or quality of service. some social concepts have various new meanings. For example, before its use in service level agreements, the notion of “contract” was introduced in software engineering in Meyer’s design by contract, a well known software design methodology that views software construction as based on contracts between clients (callers) and suppliers (routines), assertions, that has been developed in the context of object oriented and the basis of the programming language Eiffel. "Coordination" is emerging as an interdisciplinary concept to deal with the complexity of compositionality and interaction, and has been used from coordination languages in software engineering to a general interaction concept in multiagent systems. In the context of information security and access control "roles" became popular, with the popularity of eBay, the social concepts of "trust" and "reputation" have become popular, and with the emergence of social interaction sites like FaceBook or Second Life, new social concepts like societies, coalitions, organizations, institutions, norms, power, and trust are emerging [15]. In multiagent systems, social ability as the interaction with other agents and co-operation is one of the three meanings of flexibility in flexible autonomous action in Wooldridge and Jennings' weak notion of agency [46]; the other two are reactivity as interaction with the environment, and proactiveness as taking the initiative.

The main open question is whether "norms" could (or should) play a similar role in computer science like "service", "contract" or "trust"? One suggestion comes from human computer interaction. Since the use of norms is a key element of human social intelligence, norms may be essential too for artificial agents that collaborate with humans, or that are to display behavior comparable to human intelligent behavior. By integrating norms and individual intelligence normative multiagent systems provide a promising model for human and artificial agent cooperation and co-ordination, group decision making, multiagent organizations, regulated societies, electronic institutions, secure multiagent systems, and so on. Another suggestion comes from the shared interest of multiagent system research and sociology in the relation between micro-level agent behaviour and macro-level system effects. Norms are thought to ensure efficiency at the level of the multiagent system whilst respecting individual autonomy. However, all these and other suggestions bring circumstantial evidence at best. We have to build more flexible normative multiagent systems, and test them in practice, before we know where they can be used best.

For further reading on the use of normative systems in computer science, we recommend the proceedings of the ΔEON conferences and the normative multiagent systems workshops. The abstracts of all papers that appeared at DLCS conferences can be searched on the deontic logic website:

`http:\\deonticlogic.org`

References

1. C. Alchourrón and E. Bulygin. *Normative Systems.* Springer, Wien, 1971.
2. C. Alchourrón, P. Gärdenfors, and D. Makinson. On the logic of theory change. *Journal of Symbolic Logic*, 50(2):510–530, 1985.

3. A. Anderson. A reduction of deontic logic to alethic modal logic. *Mind*, 67:100–103, 1958.
4. M. Anderson and S. Leigh Anderson. Machine ethics: Creating an ethical intelligent agent. *AI Magazine*, 28(4):15–26, 2007.
5. G. Boella, J. Broersen, and L. van der Torre. Reasoning about constitutive norms, counts-as conditionals, institutions, deadlines and violations. In *Intelligent Agents and Multi-Agent Systems, 11th Pacific Rim International Conference on Multi-Agents, PRIMA 2008, Hanoi, Vietnam, December 15-16, 2008. Proceedings*, volume 5357 of *Lecture Notes in Computer Science*, pages 86–97. Springer, 2008.
6. G. Boella and L. van der Torre. The ontological properties of social roles in multi-agent systems: Definitional dependence, powers and roles playing roles. *Artificial Intelligence and Law Journal (AILaw)*, 2007.
7. G. Boella and L. van der Torre. Substantive and procedural norms in normative multiagent systems. *Journal of Applied Logic*, 6(2):152–171, 2008.
8. G. Boella, L. van der Torre, and H. Verhagen. Introduction to normative multiagent systems. *Computation and Mathematical Organizational Theory, special issue on normative multiagent systems*, 12(2-3):71–79, 2006.
9. G. Boella, L. van der Torre, and H. Verhagen. Normative multi-agent systems. In *Internationales Begegnungs und Porschungszentrum fur Informatik (IBFI)*, 2007.
10. G. Boella, H. Verhagen, and L. van der Torre. Introduction to the special issue on normative multiagent systems. *Journal of Autonomous Agents and Multi Agent Systems*, 17(1):1–10, 2008.
11. M.E. Bratman. *Intentions, plans, and practical reason*. Harvard University Press, Harvard (Massachusetts), 1987.
12. J. Broersen and L. van der Torre. Reasoning about norms, obligations, time and agents. In *Intelligent Agents and Multi-Agent Systems, 10th Pacific Rim International Conference on Multi-Agents, PRIMA 2007, Proceedings*, Lecture Notes in Computer Science. Springer, 2007.
13. E. Bulygin. Permissive norms and normative systems. In A. Martino and F. Socci Natali, editors, *Automated Analysis of Legal Texts*, pages 211–218. Publishing Company, Amsterdam, 1986.
14. C. Castelfranchi. Modeling social action for AI agents. *Artificial Intelligence*, 103(1-2):157–182, 1998.
15. C. Castelfranchi. The micro-macro constitution of power. *Protosociology*, 18:208–269, 2003.
16. R.M. Chisholm. Contrary-to-duty imperatives and deontic logic. *Analysis*, 24:33–36, 1963.
17. R. Conte, C. Castelfranchi, and F. Dignum. Autonomous norm-acceptance. In *Intelligent Agents V (ATAL98)*, LNAI 1555, pages 319–333. Springer, 1999.
18. A. Giddens. *The Constitution of Society*. University of California Press, 1984.
19. P. J. Gmytrasiewicz and E. H. Durfee. Formalization of recursive modeling. In *Procs. of ICMAS'95*, pages 125–132, Cambridge (MA), 1995. AAAI/MIT Press.
20. U. Gneezy and A. Rustichini. A fine is a price. *The Journal of Legal Studies*, 29(1):1–18, 2000.
21. L. Goble and J.J. Ch. Meyer, editors. *Deontic Logic and Artificial Normative Systems, 8th International Workshop on Deontic Logic in Computer Science, DEON 2006, Utrecht, The Netherlands, July 12-14, 2006, Proceedings*, volume 4048 of *Lecture Notes in Computer Science*. Springer, 2006.
22. L. Goble and J.J. Ch. Meyer. Revised versions of papers presented in the proceeding of the eighth international workshop on deontic logic in computer science (DEON06). *Journal of Applied Logic*, 6(2), 2008.

23. E. Goffman. *The Presentation of Self in Everyday Life.* Doubleday, 1959.
24. D. Grossi, J.-J.Ch. Meyer, and F. Dignum. Counts-as: Classification or constitution? an answer using modal logic. In *Procs. of Deontic Logic and Artificial Normative Systems, 8th International Workshop on Deontic Logic in Computer Science, (ΔEON'06)*, volume 4048 of *LNCS*, pages 115–130, Berlin, 2006. Springer.
25. J. Hansen. *Imperatives and Deontic Logic.* PhD thesis, University of Leipzig, 2008.
26. J. Hansen, G. Pigozzi, and L. van der Torre. Ten philosophical problems in deontic logic. In G. Boella, L. van der Torre, and H. Verhagen, editors, *Normative Multi-agent Systems*, volume 07122 of *Dagstuhl Seminar Proceedings*. Internationales Begegnungs- und Forschungszentrum für Informatik (IBFI), Schloss Dagstuhl, Germany, 2007.
27. B. Hansson. An analysis of some deontic logics. *Noûs*, 3:373–398, 1969.
28. M. Hollis. *Trust within reason.* Cambridge University Press, Cambridge, 1998.
29. Steven D. Levitt and Stephen J. Dubner. *Freakonomics : A Rogue Economist Explores the Hidden Side of Everything.* William Morrow, New York, May 2005.
30. D. Lewis. A problem about permission. In E. Saarinen, editor, *Essays in Honour of Jaakko Hintikka*, pages 163–175. D. Reidel, Dordrecht, 1979.
31. A. Lomuscio and D. Nute, editors. *Deontic Logic in Computer Science, 7th International Workshop on Deontic Logic in Computer Science, DEON 2004, Madeira, Portugal, May 26-28, 2004. Proceedings*, volume 3065 of *Lecture Notes in Computer Science*. Springer, 2004.
32. A. Lomuscio and D. Nute. Revised versions of papers presented in the proceeding of the seventh international workshop on deontic logic in computer science (DEON04). *Journal of Applied Logic*, 3(3-4), 2005.
33. M. Luck, P. McBurney, and C. Preist. *Agent Technology: Enabling Next Generation Computing (A Roadmap for Agent Based Computing).* AgentLink, 2003.
34. D. Makinson and L. van der Torre. Input-output logics. *Journal of Philosophical Logic*, 29(4):383–408, 2000.
35. Merriam-Webster. *Online dictionary http://www.merriam-webster.com/.* Merriam-Webster.
36. J.-J. Meyer and R. Wieringa. *Deontic Logic in Computer Science: Normative System Specification.* John Wiley & Sons, Chichester, England, 1993.
37. A. Newell. The knowledge level. *Artificial Intelligence*, 18:87–127, 1982.
38. A. Ross. Imperatives and logic. *Theoria*, 7:53–71, 1941. Reprinted in *Philosophy of Science* **11**:30–46, 1944.
39. D.W.P. Ruiter. A basic classification of legal institutions. *Ratio Juris*, 10(4):357–371, 1997.
40. J.R. Searle. *The Construction of Social Reality.* The Free Press, New York, 1995.
41. Y. Shoham and M. Tennenholtz. On social laws for artificial agent societies: Off-line design. *Artificial Intelligence*, 73(1-2):231–252, 1995.
42. A. Strauss. *Negotiations: Varieties, Contexts, Processes and Social Order.* San Francisco, Jossey-Bass, 1978.
43. R. van der Meyden and L. van der Torre, editors. *Deontic Logic in Computer Science, 9th International Conference on Deontic Logic in Computer Science, DEON 2008, Luxembourg, July 16-18, 2008, Proceedings*, LNCS, Berlin, in press. Springer.
44. L. van der Torre and Y. Tan. Diagnosis and decision making in normative reasoning. *Artificial Intelligence and Law*, 7(1):51–67, 1999.
45. G. H. von Wright. Deontic logic. *Mind*, 60:1–15, 1951.
46. M. J. Wooldridge and N. R. Jennings. Intelligent agents: Theory and practice. *Knowledge Engineering Review*, 10(2):115–152, 1995.

A categorization of simulation works on norms

Bastin Tony Roy Savarimuthu and Stephen Cranefield

Department of Information Science, University of Otago, Dunedin, P O Box 56,
Dunedin, New Zealand (tonyr,scranefield)@infoscience.otago.ac.nz

Abstract. In multi-agent systems, software agents are modelled to possess characteristics and behaviour borrowed from human societies. Norms are expectations of behaviours of the agents in a society. Norms can be established in a society in different ways. In human societies, there are several types of norms such as moral norms, social norms and legal norms (laws). In artificial agent societies, the designers can impose these norms on the agents. Being autonomous, agents might not always follow the norms. Monitoring and controlling mechanisms should be in place to enforce norms. As the agents are autonomous, they themselves can evolve new norms while adapting to changing needs. In order to design and develop robust artificial agent societies, it is important to understand different approaches proposed by researchers by which norms can spread and emerge within agent societies. This paper makes two contributions to the study of norms. Firstly, based on the simulation works on norms, we propose a life-cycle model for norms. Secondly, we discuss different mechanisms used by researchers to study norm creation, spreading, enforcement and emergence.

1 Introduction

In human societies, norms have played an important role in governing behaviour of the individuals in a society. Norms are the societal rules that govern the prescription and proscription of certain behaviour . Norms improve cooperation [1] and coordination among agents [2]. Norms reduce the amount of computation required by the agents [3] as the agents do not have to search their entire state space if they were to follow norms.

Artificial agent societies are societies in a networked environment where various agents share a virtual space and perform certain actions in a particular context (e.g. auctions). These agent societies are modelled using some of the social constructs borrowed from the human society. There have been two approaches for building normative behaviour in an agent. The first approach is the prescriptive approach where an institutional mechanism specifies how the agents should behave. The second approach is the bottom-up approach that could be used in open environments by employing mechanisms that can help a norm to emerge and govern the behaviour of an agent.

The advent of digital virtual environments such as Second Life [4] call for a distributed approach to norm spreading and emergence. Centralized policing

Dagstuhl Seminar Proceedings 09121
Normative Multi-Agent Systems
http://drops.dagstuhl.de/opus/volltexte/2009/1905

mechanism for such digital societies would be expensive from the view point of computation required due to the explosion of the states of the agents. It is impossible to monitor and control millions of agents assuming numerous roles through a centralized enforcer. A distributed approach to norms addresses these problems. Both of these approaches have been addressed by researchers and the current works focus on the issues associated with the distributed approach.

A "norm capable" agent society is the one that is able to generate, distribute, enforce and modify norms. Building robust agent societies that can create and evolve norms is important because the framework that helps in recognizing these norms will also be helpful for the agents to dynamically change these norms if situations warrant it. A good approach to test models of norm capable societies are simulations. So, a first step towards building such norm capable societies is to understand the existing simulation works on norms. To that extent, based on the simulation works on norms, we propose a life-cycle model for norms in the first part of the paper and in the second part of the paper we categorize the research works on norms based on the mechanisms employed by each of works.

2 What are norms?

Norms are expectations of an agent about the behaviour of other agents in the society. The human society follows norms such as tipping in restaurants and exchange of gifts at Christmas. Norms have been so much a part of different cultures, it is not surprising that it is an active area of research in a variety of fields including Sociology, Economics, Biology and Computer Science. Social norms have been of interest to multi-agent researchers since the early nineties. Norms are of interest to multi-agent system (MAS) researchers as they help in sustaining social order and increase the predictability of behaviour in the society. However, software agents tend to deviate from these norms due to their autonomy. So, the study of norms has become crucial to MAS researchers as they can build robust multi-agent systems using the concept of norms and also experiment with how norms evolve and adapt in response to environmental changes.

Due to multi-disciplinary interest in norms, several definitions for norms exist. Habermas [5], a renowned sociologist, identified norm-regulated actions as one of the four action patterns in human behaviour. A norm to him means *fulfilling a generalized expectation of behaviour*, which is a widely accepted definition for social norms. Ullman-Margalit [6] describes a social norm as a prescribed guide for conduct or action which is generally complied with by the members of the society. She states that norms are the resultant of complex patterns of behaviour of a large number of people over a protracted period of time. Coleman [7] describes *"I will say that a norm concerning a specific action exists when the socially defined right to control the action is held not by the actor but by others"*. Elster notes the following about social norms [1]. *"For norms to be social, they must be shared by other people and partly sustained by their approval and disapproval. They are sustained by the feelings of embarrassment,anxiety, guilt and shame that a person suffers at the prospect of violating them. A person obeying* *a norm may also be propelled by positive emotions like anger and indignation ... social norms have a grip on the mind that is due to the strong emotions they can trigger".*

Researchers have divided norms into different categories. Tuomela [8] has grouped norms into the following categories.

- r-norms (rule norms)
- s-norms (social norms)
- m-norms (moral norms)
- p-norms (prudential norms)

Rule norms are imposed by an authority based on an agreement between the members (e.g. one has to pay taxes). Social norms apply to large groups such as a whole society (e.g. one should not litter). Moral norms appeal to one's conscience (e.g. one should not steal or accept bribes). Prudential norms are based on rationality (e.g one ought to maximize one's expected utility). When members of a society violate the societal norms, they may be punished.

Many social scientists have studied why norms are adhered to. Some of the reasons for norm adherence include:

- fear of authority or power
- rational appeal of the norms
- emotions such as shame, guilt and embarrassment that arise because of non-adherence.
- willingness to follow the crowd

Elster [1] categorizes norms into consumption norms (e.g. manners of dress), behaviour norms (e.g. the norm against cannibalism), norms of reciprocity (e.g. gift-giving norms), norms of cooperation (e.g. voting and tax compliance) etc.

For the purpose of this paper, we focus on social norms because the agents in multi-agent systems have been modelled using ideas borrowed from sociology such as speech act theory and autonomy. Software agents are the proxies for human agents and possess these human-like attributes. Agents acting on behalf of humans (e.g. in virtual worlds) or as independent entities (bots) will need this notion of social norms that regulate their behaviour. Based on the definitions provided by various researchers, we note that the notion of a norm is generally made up of the following two aspects.

- Normative expectation of a behavioural regularity: There is a general agreement within the society that a behaviour is expected on the part of an agent (or actor) by others in a society, in a given circumstance.
- A norm spreading factor : Examples of norm spreading factors include the notion of advice from powerful leaders and the sanctioning mechanism. When an agent does not follow the norm, it could be subjected to a sanction. The sanction could include monetary or physical punishment in the real world which can trigger emotions (embarrassment, guilt etc.) or direct loss of utility. Other kind of sanctions could include agents not being willing to interact with an agent that violated the norm or the decrease of its reputation

score. Other norm spreading factors include imitation and learning on the part of an agent.

It should be noted that researchers are divided on what the differences between a norm and a convention are. Our belief is that convention is a common expectation amongst (most) others that an agent adopts a particular action or behaviour (e.g. the convention in ancient Rome was to drive on the left). In this paper we do not distinguish conventions from norms. Both of them have been incorporated under the umbrella of *norms*.

2.1 Normative multi-agent systems

Research on norms in multi-agent systems is about two decades old. [9–11]. Norms in multi-agent systems are treated as constraints on behaviour, goals to be achieved or as obligations [12].

The definition of normative multi-agent systems as described by the researchers involved in the NorMAS 2007 workshop is as follows [13]. *A normative multi-agent system is a multi-agent system organized by means of mechanism to represent, communicate, distribute, detect, create, modify and enforce norms, and mechanisms to deliberate about norms and detect norm violation and fulfillment.*

The research in normative multi-agent systems can be categorized into two branches. The first branch focuses on normative system architectures, norm representations, norm adherence and the associated punitive or incentive measures. Lopez et al. [14] have designed an architecture for normative BDI agents and [15] have proposed a distributed architecture for normative agents. Some researchers are working on using deontic logic to define and represent norms [15, 16]. Several researchers have worked on mechanisms for norm compliance and enforcement [17–19]. A recent development is the research on emotion based mechanisms for norm enforcement [20, 21]. Conte and Castelfranchi [22] have worked on an integrated view of norms. Their work tries to bridge the gap between the prescriptive view of norms and the emergence of conventions from mere regularities using cognitive abilities of an agent. For a comparison of normative architectures refer to Neumann's article [23].

The second branch of research is related to emergence of norms. Neumann has presented a case study of four research works on the simulation models of norms from the perspective of foundations of social theory [24]. In this work, the four papers were investigated in detail for identifying three methodological core problems which are norm transmission, norm transformation and the function of the norm. The first two problems correspond to the causal aspect of the norm (i.e. what causes the norm to spread). The last problem deals with the purpose of the norm. The author concludes that no model has been able to fully explain both the causal and functional reasons behind norm emergence, however, the current trend is towards bridging this gap.

3 Phases of norm life-cycle

Researchers interested in norms have experimented with several mechanisms associated with norms. Firstly we identify four phases of the norm life-cycle. Secondly, we categorize the simulation mechanisms into 10 categories and have assigned each category to a particular phase of the norm life-cycle.

In the body of research literature on social norms there isn't a unified view on how norms are created and spread in a society. Several researchers have proposed models of norms [1, 7, 25–28]. In this paper we refer to four important phases of norm life-cycle which are norm creation, spreading, enforcement and emergence. Even though there hasn't been any agreement on these phases by social researchers, we use these four phases as they broadly capture the processes associated with the norm life-cycle. Figure 1 shows the four phases of norm life-cycle (in the left) and the categories of mechanisms (in the right).

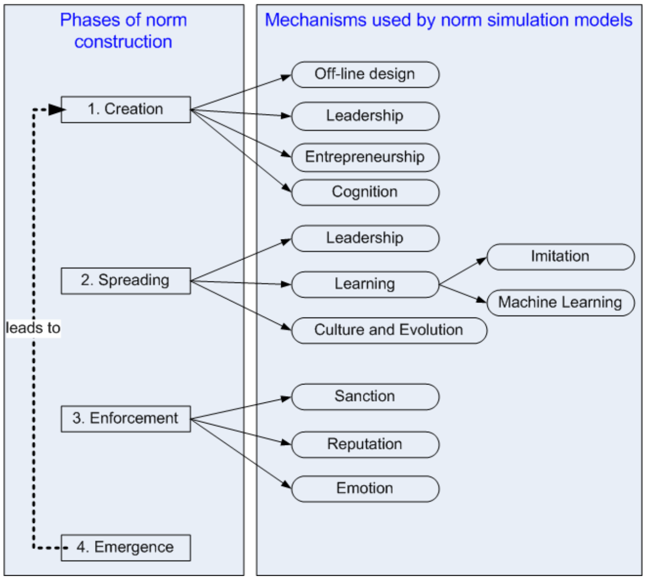


Fig. 1: Phases of norm life-cycle and categories of simulation models

The first phase of the life-cycle model is that of norm creation. A norm can be created by a designer of the system or a powerful leader. The designer and leadership approaches are top-down authoritarian approaches. The other approach for norm creation is the entrepreneurial approach where an agent might come up with a norm and can recommend the norm to other agents. These norms when created are the ”proposed norms”. Once such a proposed norm is created by a designer, leader or entrepreneur, it spreads through the society by one of the spreading mechanisms such as advice about a norm by powerful members of the society, imitation, learning on the part of the agents, cultural inheritance and evolutionary inheritance. Thus, norm spreading forms the second phase of the norm life-cycle. When norms have spread and are internalized, agents may expect other agents to follow the norm that they have subscribed to and may sanction those agents that do not follow the norm. The third phase of the life-cycle is the enforcement of norms where norm violators may be punished, their utility might be reduced, their reputation impacted or emotions such as shame and guilt being stirred which help in the regulation of normative behaviour. The fourth phase is the norm emergence phase. A norm can be said to have emerged if it has spread (i.e. it is followed by a considerable proportion of an agent society and this fact is recognized by most agents). Another aspect of norm emergence is that a norm can emerge without being explicitly created. Norms can emerge in a bottom-up way. One or more cognitive agents, based on interactions in an agent society can infer what the norms of the society are. We can say that these agents derived a "proposed norm" based on their cognitive ability (creation phase) and then helped in the emergence of that norm (emergence phase). These cognitive agents can also come up with an alternative norm that spreads and emerges in a society and hence can replace an existing norm. This feedback loop is represented as a dashed line in 1. It should be noted that not all simulation based research on norms have considered all the 4 phases.

The life-cycle that we have presented is similar to Finnemore and Sikkink's model [27]. They have proposed three stages for the norm life cycle. The first stage is the norm emergence stage which is characterized by the persuasion by some norm entrepreneurs or norm innovators. Norm entrepreneurs are the innovators who think about new ideas/norms in a society. Norm entrepreneurs attempt to convince a critical mass of norm leaders to embrace new norms. The second stage is characterized by the dynamics of imitation as the norm leaders attempt to socialize with other people whom they might have influence over, so they might become followers. The third stage is the norm cascade stage where the followers take up the norm for reasons such as pressure to conform. As the reader may observe, this model is a subset of the life-cycle model that we have proposed. Also, this model caters for the entrepreneurial approach for norm creation and the imitation approach for norm spreading. But, in our approach, more mechanisms are brought under each of the phases.

3.1 Norm creation

Norm creation in multi-agent system refers to the mechanism by which an agent in the society comes to know what the norm of the society is. There are three approaches that simulation works have used and they are a) a designer specified norms (off-line design) b) a norm-leader specified norms and c) a norm-entrepreneur considers that a norm is good for the society.

Off-line design approach - In this approach, norms are designed off-line, and hardwired into agents. Walker and Wooldridge [29] note the following about the off-line design of norms. *"The off line design of norms will often be simpler* *to implement and might present the designer with a greater degree of control over system functionality. However, there are a number of disadvantages with this approach. First, it is not always the case that all the characteristics of a system are known at design time; this is most obviously true of open systems Secondly, in complex system the goals of agents might be constantly changing. To keep reprogramming agents in such circumstances would be costly and inefficient. Finally, the more complex the a system becomes, the less likely it is that system designers will be able to design effective social laws".*

Some researchers have used this approach to compare the performance of a normative system with a non normative one [30].This approach is only suitable for top-down norm prescription that is a characteristic of closed and centralized institutions.

Leadership approach - In this approach, some powerful agents in the society (the norm-leaders) come up with a norm. The leader can provide these norms to the follower agents [10, 31].

Entrepreneurial approach - In agent societies, there might be some norm-entrepreneurs who come up with a norm. When an agent comes up with a new norm it tries to convince other agents [32].

Cognitive approach - One or more cognitive agents in a society can come up with norms based on the deliberative processes that they employ [33]. In this approach the agents have the cognitive ability to recognize what the norms of a society are based on the observations of interactions. It should be noted that the norm inferred by each agent might be different (which is based on the observations that an agent has made). Thus, an agent in this model creates its notion of what the norm is based on inference.

3.2 Norm spreading

Norm spreading relates to the distribution of a norm among a group. Mirriam-Webster's dictionary [34] defines spreading as *to become dispersed, distributed, or scattered or to become known or disseminated.* There are several mechanisms that help in spreading the norms such as leadership, imitation , machine learning, cultural and evolutionary mechanisms. These mechanisms are discussed in detail in the next section.

3.3 Norm enforcement

Norm enforcement refers to the process by which norm violators are discouraged through some form of sanctioning. A widely used sanctioning mechanism is the punishment of a norm violator (e.g. monetary punishment which reduces the agents fitness or a punishment that invokes emotions such as guilt and embarrassment). Reputation mechanisms have also been used as sanctions where an agent is black-listed for not following a norm. The process of enforcement helps to sustain norms in a society. Some researchers have considered enforcement as a part of the spreading mechanism [19].

3.4 Norm emergence

We define norm emergence to be reaching some significant threshold in the extent of the spread of a norm. For example, a society is said to have a norm of gift exchange at Christmas if more than $x\%$ of the population follows such a practice. The value of x varies from society to society and from one kind of norm to another. The value of x has varied from 35 to 100 across different simulation based studies of norms.

Simulation research on norms has employed two approaches to norm emergence. One approach is that an agent comes to know about a norm through mechanisms such as leadership [31, 35] or through imitation [3] and when it accepts the norm it contributes to norm spreading and emergence. The other way is that a cognitive agent could generate a personal norm based on observation [36]. Additionally many such cognitive agents in the society could generate similar personal norms and for an external observer it might seem that a norm has emerged in a society. Also, cognitive agents could communicate norms and verify norms. The later bottom-up approach where micro interactions between agents that lead to the macro effect of establishing a norm is more interesting than the leadership and imitation based approaches.

4 Categorization of simulation works of norm creation, spreading and emergence

In this section we categorize simulation work on norms into eight main categories (shown in figure 2). Each category corresponds to a particular mechanism (e.g. sanctioning mechanism, reputation mechanism). For each of these categories we provide a brief description and discuss a few key papers. It should be noted that some papers have made use of mechanisms that fall under more than one category.

4.1 Social power

Social power plays an important role in societies in establishing order and enabling smoother functioning. Several researchers in normative multi-agent systems have focused on the notion of power [37–39] such as institutional power. Lopez in her thesis on social power and norms notes that powers of an agent are expressed through its abilities to change the beliefs, the motivations, and the goals of other agents in such a way that its goals can be satisfied [40].

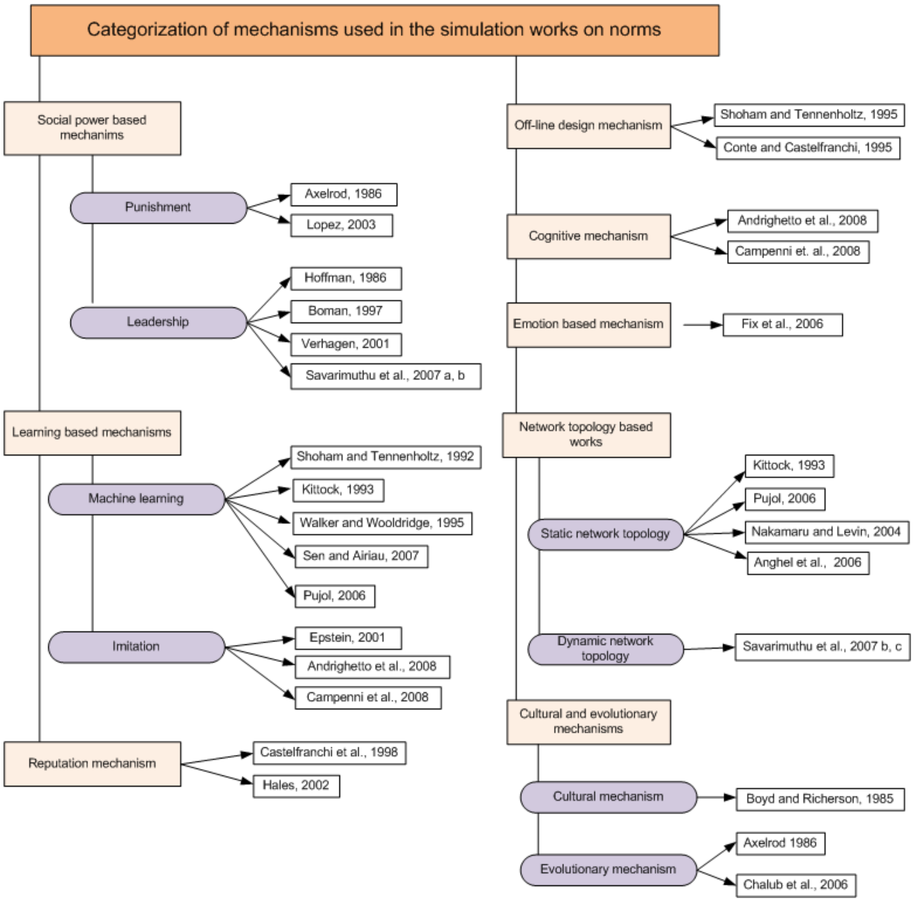


Fig. 2: Categorization of simulation models

Sources of power could motivate, encourage or coerce their followers to take up a particular norm (leadership approach) or force them to adopt a particular norm based on sanctions (punishment approach). Researchers have experimented with both types of social power approaches for norm spreading and enforcement.

Leadership mechanism - Leadership mechanisms are based on the notion that there are certain leaders in the society. These leaders provide advice to the agents in the society. The follower agents seek the leaders advice when deciding about a norm. Verhagen [31] has used the concept of normative advice (advice from the leader of a society) as one of the mechanisms for spreading and internalizing norms in an agent society. However, this centralized approach might not work well in open, flexible and dynamic societies. Savarimuthu et al. [35] extended Verhagen's model by adopting a distributed mechanism for norm emergence. In their mechanism, there could be several normative advisors or role models whom other agents can request for advice. In this model, an agent can be a leader for some agents while that agent itself can be a follower of some other agent. Hoffmann [32] has experimented with the notion of norm entrepreneurs who think of a norm that might be beneficial to the society. His experiments explore the entrepreneurial norm dynamics and provide some initial evidence for Finnemore and Sikkink's norm life cycle model [27].

Sanction mechanism - Even though the models discussed above are based on the notion of power and leadership, they do not include the notion of sanctioning agents that do not follow the norm specified by a norm leader. Several works on norms have used the notion of social power to inflict sanctions on agents that do not follow a norm [17,19,41]. In his well known work [19], Axelrod has shown how a meta-norm that defections should be punished can bring about the norm of cooperation. Lopez et al. [17] have considered punishments and rewards in their model. Their framework models agents with different personalities (social, pressured, opportunistic, greedy, fearful, rebellious). A proper account of the cost of punishment has not been considered in both these works. While Axelrod's work does not consider cost of punishment on the part of the sanctioning agent, Lopez's work assumes that a third party somehow bears this cost. Flentge et al. [41] have shown how an agent comes to acquire a possession norm. They have noted that sanctions help in the establishment of the possession norm if the sanctioning costs are low or when there is no cost for sanctioning.

4.2 Reputation mechanism

Reputation refers to the positive or negative opinion about a person or agent based on their interactions with others in the society. Researchers [42,43] have addressed how reputation models are beneficial in sustaining norms in an agent society. They have experimented with the effect of the normative reputation on the compliance costs of the norm. They have shown that the normative reputation of agents of the society helps in redistributing the costs of norm compliance to both the agents that follow the norms as well as those who do not follow the norms.

4.3 Imitation mechanism

The philosophy behind an imitation mechanism is *When in Rome, do as Romans do* [3]. These models are characterized by agents mimicking the behaviour of what the majority of the agents do in a given agent society (following the crowd). Epstein's main argument for an imitation mechanism is that individual thought (i.e. the amount of computing needed by a agent to infer what the norm is) is inversely related to the strength of a social norm [3]. This implies that when a norm becomes entrenched the agent can follow it without much thought. Epstein has demonstrated this in the context of a driving scenario in which agents can observe each other's driving preference (left or right) based on a certain observation radius r. If the agent sees more agents driving on the right within the observation radius,it changes to the right. When a norm is established, the radius tends to move towards one. Other researchers have also experimented with imitation models [36,40,44]. This might be a good mechanism when agents want to avoid the cost of thinking about what the norm of the society is. An agent using the imitation model is not involved in the creation of the norm, it is just a part of the norm spreading effort. Though simple, the model can only account for a way to spread the norm (which is blindly following it). It has been noted that imitation approach cannot bring about the co-existence of multiple norms in a society [45, 46]. Also, it is debatable if imitation-based behaviour (solely) really leads to norms as there is no notion of common expectation.

4.4 Off-line design approach

Off-line design models are characterized by the agents of the society possessing explicit knowledge of the norm. The intention of the designer specified approach is to see how the society performs when the whole society possesses a norm. One of the well-known works on norms specified by the designer is Shoham and Tennenholtz [9]. They have experimented with norms associated with traffic. Several other researchers [29, 43, 47] have experimented with an off-line design approach borrowing the basic experimental set-up proposed by Conte and Castelfranchi [47]. Conte and Castelfranchi have shown using their simulation experiments what the function of a norm is in the context of agents finding food in a grid environment characterized by simple rules for movement and food collection. They have compared the utilitarian strategy with the normative strategy. They have shown that norms reduce the aggression level of the agent (i.e. when a finder-keeper norm is followed) and also increase the average strength of an agent.

4.5 Works based on machine learning

Several researchers have experimented with agents finding a norm based on learning on the part of an agent [2, 29, 48]. Shoham and Tennenholtz have used a mechanism called co-learning which is a simple reinforcement learning mechanism. They have used the "Highest Cumulative Reward (HCR)" rule to update an agent's strategy when playing a simple coordination game and a cooperation game (prisoner's dilemma). According to this rule, an agent chooses the strategy that has yielded the highest reward in the past m iterations. The history of the strategies chosen and the rewards for each strategy is stored in a memory of a certain size (which can be varied). Walker and Wooldridge's experimental model [29] is based on the work done by Conte and Castelfranchi [47] where agents move about a grid in search of food. They have experimented with 16 mechanisms for norm emergence. Their model used two parameters, the majority and the strategic update function. Each of these parameters can be varied with four values. 16 experiments were based on size of the majority (simple, double, quadruple, dynamic) and the nature of the update function (using majority rule, memory restart, communication type and communication on success). Sen and Airiau [48] have proposed a mechanism for the emergence of norms through social learning. They have experimented with three reinforcement learning algorithms and the agents learn norms based on private local interactions. They have observed that when the population size is bigger the norm convergence is slower and larger the set of possible action states the slower is the convergence. They have also studied the influence of adding agents with a particular action state to a pool of existing agents as well as norm emergence in isolated sub-populations.

Learning mechanisms employ a particular algorithm to identify a strategy that maximizes an agent's utility and the chosen strategy is declared as the norm. Since all agents in the society make use of the same algorithm, the society stabilizes to a uniform norm. Agents using this approach cannot distinguish between a strategy and a norm. The agents do not have a notion of normative expectation (i.e. others expect certain behaviour on the part of an agent) associated with a norm.

4.6 Cognitive approach

Researchers involved in the EMIL project [33] are working on a cognitive architecture for norm emergence. There have been some attempts to explore how the mental capacities of agents play a role in the emergence of norms.

EMIL project aims to deliver a simulation-based theory of norm-innovation, where norm-innovation is defined as a 2-way dynamics of inter-agent process and intra-agent process. The inter-agent process results in the emergence of norms where the micro interactions produce macro behaviour (norms). The intra-agent process refers to what goes inside an agent's mind so that they can recognize what the norms of the society are. This approach is different from the learning models as the agents in the cognitive approach are autonomous and have the capability to examine interactions between agents and are able to recognize what the norms could be. The agents in this model need not necessarily be utility maximizing like the ones in the learning models. The agents in the model will have the ability to filter external requests that affect normative decisions and will also be able to communicate norms with other agents. Agents just employing learning algorithms lack these capabilities.

Andrighetto et al. [36] have demonstrated how the norm recognition module of the EMIL-A platform answers the question "how does a agent come to know of what a norm is". In particular they have experimented with an imitation approach versus the norm recognition approach that they have come up with. The norm recognition module consists of two constructs, the normative board and a module for storing different types of modals for norms. Each modal represents a type of message that is exchanged between agents (e.g. deontics modal refers to partitioning situations as either acceptable or unacceptable). The normative board consists of normative beliefs and normative goals. They have shown that norm recognizers perform better than social conformers (imitating agents) by the fact that the recognizers were able to identify a pool of potential norms while the imitators generated only one type of norm.

The limitation of this approach is that agents just observe actions performed by other agents. In practice they should be able to learn from their own experience as well. Perhaps, their own experience can be given a higher weight. At present, agents in their model do not have the capability of violating the norms and hence there are no costs associated with sanctions. The authors note this can be a potential extension.

4.7 Emotion based works

Based on the previous work done by Scheve et al. [20], Fix et al. [21] discuss the micro-macro linkage between emotions at the micro-level and the norm enforcement at the macro-level. The authors argue that emotions have a norm regulatory function in agent societies. An agent observing a deviation of a norm might generate emotions such as contempt or disgust which can be the motivation behind sanctions. Those agents that are sanctioned might generate emotions such as shame, guilt or embarrassment which might lead to norm internalization. The authors have used a Petri net model [49] to capture the micro-macro linkage. It should be noted that the proposed model has not been implemented in the context of a simulation experiment. Staller and Petta [50] have extended Conte et al.'s experimental set up by including emotion based strategies.

4.8 Works using network topologies

Social networks are important for norm spreading and emergence because in the real world, people are not related to each other by chance. They are related to each other through the social groups that they are in, such as the work group, church group, ethnic group and hobby group. Information tends to percolate among the members of the group through interactions. Also, people seek advice from a close group of friends and hence information gets transmitted between the members of the social network.

In most simulation works, the treatment of norms has been mostly in the context of an agent society where the agents interact with all the other agents in the society [10, 31] in a random fashion. Few researchers have considered the actual topologies of the social network for norm emergence [44]. We believe such an approach is important for the study of norm spreading and emergence as networks provide the topology and the infrastructure on which the norms can be exchanged. Researchers have studied different kinds of network topologies and their applications in the real world (a overview of different topologies is given by Mitchell [51]). These application areas include opinion dynamics [52] and the spread of diseases [53]. Researchers in normative multi-agent systems have started to look at the role of network topologies [44, 54–56]. Network topologies have also been explored by other multi-agent system researchers in other contexts such as reputation management [57, 58].

Research that has considered network topologies can be categorized into static and dynamic network topology approaches. In the static approach, the network topology is fixed. In the dynamic topology approach, the underlying network can change when the simulation experiments are conducted.

Works using a static network topology - Kittock was the first to experiment with the role of network topology in convention emergence [54]. He noted that the choice of the global structure has a profound effect on the evolution of the system. Pujol's PhD thesis [44] dealt with the emergence of conventions on top of social structures. He used the HCR mechanism proposed by Shoham and Tennenholtz [2] to test norm emergence in connected, random, small world and scale-free networks. He also demonstrated that the structure of the network is crucial for norm emergence. Nakamaru and Levin [46] studied how two related norms evolve in networked environments. Anghel et al. [59] investigated the effects of inter-agent communication across a network in the context of playing minority game. They have shown that a scale-free leadership structure emerges on top of a random network.

Dynamic topology works - Very few researchers have investigated the role of dynamic network topologies on norm spreading and emergence. Savarimuthu et al. [55] used Gonzalez et al.'s model [60] to create dynamic network topologies. Gonzalez et al. have developed a model for constructing dynamically changing networks. They have used the concept of agents (or particles) colliding in an abstract social space to construct evolving networks. Savarimuthu et al. [55] have created dynamic network topologies using Gonzalez's model on which they test their role model agent-based leadership mechanism. They have shown how different types of norms emerge when societies with different norms for the same context (playing the Ultimatum game [61]) are brought together. In particular, they have shown that under certain conditions norms can co-exist in an agent society.

4.9 Cultural and evolutionary mechanisms

Researchers have also proposed other mechanisms for norm spreading and emergence. These include cultural and evolutionary models [62, 63]. Boyd and Richerson [62] have proposed that norms can be propagated through cultural transmission. According to them, there are three ways by which a social norm can be propagated from one member of the society to another. They are

- Vertical transmission (from parents to offspring)
- Oblique transmission (from a leader of a society to the followers)
- Horizontal transmission (from peer to peer interactions)

Of these three kinds of norm transmission mechanisms, vertical and oblique transmissions can be thought of as leadership mechanisms in which a powerful superior convinces the followers to adopt a norm. The horizontal transmission is a peer-to-peer mechanism where agents learn from day-to-day interactions from other peers. Few researchers have used this idea to experiment with norm spreading [31, 64].

A few researchers have experimented with norm spreading based on evolution where the offsprings inherit the behaviour of the parents. One well known work in this category is Axelrod's [19]. Few other researchers have also experimented with evolutionary models for norm spreading [56, 63]. Chalub et al. [63] have experimented on how norms might spread in different societies (e.g. an archipelago of islands). Agents in an island are fully connected to each other. Each agent plays the donor-receiver game once with all other agents in the island. Then an agent reproduces by choosing a connected agent at random and comparing the payoff. If its payoff is higher than the other agent, then the other agent inherits the strategy of the winning player. Each island has a Gross Domestic Product (GDP) which is a normalized average payoff of the entire island. Islands compete against each other. There are times of war and peace. During peace times, the norms of the islands do not change. When the islands are at war, they play the Hawk and Dove [65] game. The losers change their norm based on a probabilistic norm update rule. The authors note that a meta-norm is established at the end of each run. One limitation of this approach is that they assume that norms have somehow been internalized by a parent/propagator.

Table 1 shows the mechanisms used by the various simulation works on norms corresponding to each phase of the norm life-cycle. It should be noted that not all phases of norm life-cycle have been taken into account by most works.

5 Conclusions

This paper has made two contributions to normative multi-agent system field in the context of simulation of norms. Firstly, a four phase model of the norm life-cycle was proposed. Secondly, various norm-based simulation works were categorized based on the mechanisms employed by each of the works. In the future, we intend to elaborate the research that has been carried out using each of the mechanisms discussed in this paper and also compare their strengths and weaknesses. We will also compare the simulation works based on the agent characteristics employed in each of the works. We also intend to discuss the research issues that need to be addressed.

References

1. Elster, J.: Social norms and economic theory. The Journal of Economic Perspectives **3**(4) (1989) 99–117
2. Shoham, Y., Tennenholtz, M.: Emergent conventions in multi-agent systems: Initial experimental results and observations (preliminary report). In: KR. (1992) 225–231
3. Epstein, J.M.: Learning to be thoughtless: Social norms and individual computation. Comput. Econ. **18**(1) (2001) 9–24

4. : Second life. http://secondlife.com/
5. Habermas, J.: The Theory of Communicative Action : Reason and the Rationalization of Society. Volume 1. Beacon Press (1985)
6. Ullmann-Margalit, E.: The Emergence of Norms. Clarendon Press (1977)
7. Coleman, J.: Foundations of Social Theory. Belknap Press (August 1990)
8. Tuomela, R.: The Importance of Us: A Philosophical Study of Basic Social Notions. Stanford Series in Philosophy, Stanford University Press (1995)
9. Shoham, Y., Tennenholtz, M.: On social laws for artificial agent societies: Off-line design. Artificial Intelligence **73**(1-2) (1995) 231–252
10. Boman, M.: Norms in artificial decision making. Artificial Intelligence and Law **7**(1) (1999) 17–35
11. Conte, R., Falcone, R., Sartor, G.: Agents and norms: How to fill the gap? Artificial Intelligence and Law **7**(1) (1999) 1–15
12. Castelfranchi, C., Conte, R.: Cognitive and social action. UCL Press, London (1995)
13. Boella, G., Torre, L., Verhagen, H.: Introduction to the special issue on normative multiagent systems. Autonomous Agents and Multi-Agent Systems **17**(1) (2008) 1–10
14. López y López, F., Márquez, A.A.: An architecture for autonomous normative agents. In: Fifth Mexican International Conference in Computer Science (ENC'04), Los Alamitos, CA, USA, IEEE Computer Society (2004) 96–103
15. Boella, G., van der Torre, L.: An architecture of a normative system: counts-as conditionals, obligations and permissions. In: AAMAS, New York, NY, USA, ACM Press (2006) 229–231
16. García-Camino, A., Rodríguez-Aguilar, J.A., Sierra, C., Vasconcelos, W.: Norm-oriented programming of electronic institutions. In: Proceedings of the fifth international joint conference on autonomous agents and multiagent systems, AAMAS, New York, NY, USA, ACM Press (2006) 670–672
17. López y López, F., Luck, M., d'Inverno, M.: Constraining autonomy through norms. In: Proceedings of The First International Joint Conference on Autonomous Agents and Multi Agent Systems AAMAS'02. (2002) 674–681
18. Aldewereld, H., Dignum, F., García-Camino, A., Noriega, P., Rodríguez-Aguilar, J.A., Sierra, C.: Operationalisation of norms for usage in electronic institutions. In: AAMAS, New York, NY, USA, ACM Press (2006) 223–225
19. Axelrod, R.: An evolutionary approach to norms. The American Political Science Review **80**(4) (1986) 1095–1111
20. Scheve, C., Moldt, D., Fix, J., Luede, R.: My agents love to conform: Norms and emotion in the micro-macro link. Comput. Math. Organ. Theory **12**(2-3) (2006) 81–100
21. Fix, J., von Scheve, C., Moldt, D.: Emotion-based norm enforcement and maintenance in multi-agent systems: foundations and petri net modeling. In: AAMAS. (2006) 105–107
22. Conte, R., Castelfranchi, C.: From conventions to prescriptions - towards an integrated view of norms . Artif. Intell. Law **7**(4) (1999) 323–340
23. Neumann, M.: A classification of normative architectures. In: WCSS-08 Proceedings, not known (2008) not known
24. Neumann, M.: Homo socionicus: a case study of simulation models of norms. Journal of Artificial Societies and Social Simulation **11**(4) (2008) 6
25. Opp, K.D.: How do norms emerge? An outline of a theory. Mind and Society **2**(1) (2001) 101–128

26. Horne, C.: Sociological perspectives on the emergence of norms. Social Norms (Hechter, M. and Opp, KD, eds) (2001) 3–34
27. Finnemore, M., Sikkink, K.: International Norm Dynamics and Political Change. International Organization **52**(04) (2005) 887–917
28. Bicchieri, C.: The Grammar of Society - The Nature and Dynamics of Social Norms. Cambridge University Press (2006)
29. Walker, A., Wooldridge, M.: Understanding the emergence of conventions in multi-agent systems. In Lesser, V., ed.: Proceedings of the First International Conference on Multi–Agent Systems, San Francisco, CA, MIT Press (1995) 384–389
30. Conte, R., Castelfranchi, C.: Norms as mental objects - from normative beliefs to normative goals. In: MAAMAW. (1993) 186–196
31. Verhagen, H.: Norm Autonomous Agents. PhD thesis, Department of Computer Science, Stockholm University (2000)
32. Hoffmann, M.: Entrepreneurs and Norm Dynamics: An Agent-Based Model of the Norm Life Cycle. Technical report, Department of Political Science and International Relations, University of Delaware, USA (2003)
33. Andrighetto, G., Conte, R., Turrini, P., Paolucci, M.: Emergence in the loop: Simulating the two way dynamics of norm innovation. In Boella, G., van der Torre, L., Verhagen, H., eds.: Normative Multi-agent Systems. Number 07122 in Dagstuhl Seminar Proceedings, Internationales Begegnungs- und Forschungszentrum fuer Informatik (IBFI), Schloss Dagstuhl, Germany (2007)
34. : Spreading - definition from the merriam-webster online dictionary
35. Savarimuthu, B.T.R., Purvis, M.A., Cranefield, S., Purvis, M.K.: How do norms emerge in multi-agent societies? mechanisms design. (2007)
36. Andrighetto, G., Campenni, M., Cecconi, F., Conte, R.: How agents find out norms: A simulation based model of norm innovation. In: Not known, not known (2008) not known
37. Castelfranchi., C., Cesta, A., Miceli, M.: Dependence relations among autonomous agents. In: Decentralized A.I.-3. Elsevier, Amsterdam (1992)
38. Jones, A.J.I., Sergot, M.J.: A formal characterisation of institutionalised power. Logic Journal of the IGPL **4**(3) (1996) 427–443
39. Castelranchi, C.: C. Castelfranchi. All I understand about power (and something more). Technical report, ALFEBIITE Project, London (2000)
40. López y López, F.: Social Powers and Norms: Impact on Agent Behaviour. PhD thesis, University of Southampton, United Kingdom (2003)
41. Flentge, F., Polani, D., Uthmann, T.: Modelling the emergence of possession norms using memes. Journal of Artificial Societies and Social Simulation **4** (2001)
42. Castelfranchi, C., Conte, R., Paolucci, M.: Normative reputation and the costs of compliance. Journal of Artificial Societies and Social Simulation **vol. 1, no. 3** (1998)
43. Hales, D.: Group reputation supports beneficent norms. Journal of Artificial Societies and Social Simulation **5** (2002)
44. Pujol, J.M.: Structure in Artificial Societies. PhD thesis, Software Department, Universitat Politénica de Catalunya (2006)
45. Campenni, M., Andrighetto, G., Cecconi, F., Conte, R.: Normal = normative? the role of intelligent agents in norm innovation. In: Not known, not known (2008) not known
46. Nakamaru, M., Levin, S.A.: Spread of two linked social norms on complex interaction networks. Journal of Theoretical Biology **230**(1) (September 2004) 57–64

47. Conte, R., Castelfranchi, C.: Understanding the effects of norms in social groups through simulation. In: Artificial societies: the computer simulation of social life. UCL Press, London (1995)
48. Sen, S., Airiau, S.: Emergence of norms through social learning. In: Proceedings of Twentieth International Joint Conference on Artificial Intelligence (IJCAI), Hyderabad, India, MIT Press (2006) 1507–1512
49. Jensen, K., ed.: Application and Theory of Petri Nets 1992, 13th International Conference, Sheffield, UK, June 22-26, 1992, Proceedings. In Jensen, K., ed.: Application and Theory of Petri Nets. Volume 616 of Lecture Notes in Computer Science., Springer (1992)
50. Staller, A., Petta, P.: Introducing emotions into the computational study of social norms: A first evaluation. J. Artificial Societies and Social Simulation **4**(1) (2001)
51. Mitchell, M.: Complex systems: Network thinking. Artificial Intelligence **170**(18) (2006) 1194–1212
52. Fortunato, S.: Damage spreading and opinion dynamics on scale free networks (2004)
53. Cohen, R., Havlin, S., ben Avraham, D.: Efficient immunization strategies for computer networks and populations. Physical Review Letters **91** (2003) 247901
54. Kittock, J.E.: Emergent conventions and the structure of multi-agent systems. In Nadel, L., Stein, D.L., eds.: 1993 Lectures in Complex Systems. Addison-Wesley (1995)
55. Savarimuthu, B.T.R., Cranefield, S., Purvis, M.K., Purvis, M.A.: Norm emergence in agent societies formed by dynamically changing networks. In: IAT '07: Proceedings of the 2007 IEEE/WIC/ACM International Conference on Intelligent Agent Technology, Washington, DC, USA, IEEE Computer Society (2007) 464–470
56. Villatoro, D., Sabater-Mir, J.: Categorizing social norms in a simulated resource gathering society. In: Proceeding of the Advancement of Artificial Intelligence (AAAI) workshop on Coordination, Organization,Institutions and Norms in agent systems (COIN). (2008) not known
57. Pujol, J.M., Sangüesa, R., Delgado, J.: Extracting reputation in multi agent systems by means of social network topology. In: Proceedings of the first international joint conference on autonomous agents and multiagent systems, AAMAS, New York, NY, USA, ACM Press (2002) 467–474
58. Yu, B., Singh, M.P.: Searching social networks. In: Proceedings of the second international joint conference on autonomous agents and multiagent systems, AAMAS, New York, NY, USA, ACM Press (2003) 65–72
59. Anghel, M., Toroczkai, Z., Bassler, K.E., Korniss, G.: Competition-driven network dynamics: Emergence of a scale-free leadership structure and collective efficiency. Physical Review Letters **92**(5) (2004) 0587011–0587014
60. Gonzaléz, M.C., Lind, P.G., Herrmann, H.J.: Networks based on collisions among mobile agents. Physica D **224** (2006) 137–148 e-print: physics/0606023.
61. Slembeck, T.: Reputations and fairness in bargaining - experimental evidence from a repeated ultimatum game with fixed opponents. Experimental 9905001, Economics working paper archive (1999)
62. Boyd, R., Richerson, P.J.: Culture and the evolutionary process. University of Chicago Press, Chicago (1985)
63. Chalub, F., Santos, F., Pacheco, J.: The evolution of norms. Journal of Theoretical Biology **241**(2) (2006) 233 – 240
64. Savarimuthu, B.T.R., Cranefield, S., Purvis, M.A., Purvis, M.P.: Role model based mechanism for norm emergence in artificial agent societies. In: Proceeding of the

AAMAS 2007 workshop on Coordination, Organization, Institutions and Norms in agent systems (COIN). (2007) 1–12

65. Smith, M.J., Price, G.R.: The logic of animal conflict. Nature **246**(5427) (November 1973) 15–18

Simulation works	**Norm creation**	**Norm spreading**	**Norm enforcement**	**Norm emergence**
Axelrod, 1986	-	Evolutionary approach	Sanction	Yes
Shoham and Tennenholtz, 1992	Learning	-	-	Yes
Kittock, 1993	-	Learning, network topology	-	Yes
Conte and Castelfranchi, 1995	Off-line	-	-	-
Walker and Woolridge, 1995	-	Learning	-	Yes
Shoham and Tennenholtz, 1995	Off-line	-	-	-
Castelfranchi et al., 1998	Off-line	-	Reputation	-
Verhagen, 2000	Leadership	Leadership	-	-
Epstein, 2001	-	Imitation	-	Yes
Flentge et al., 2001	-	Cultural transmission	Sanction	Yes
Hales, 2002	Off-line	-	Reputation	-
Hoffmann, 2003	Entrepreneurship	leadership	-	Yes
Lopez et al., 2003	Off-line	-	Sanction and reward	-
Nakamaru and Levin, 2004	Off-line	Network topology	-	Yes
Chalub et al., 2006	-	Evolutionary approach	-	Yes
Fix et al., 2006	-	-	Emotion	-
Pujol, 2006	-	Learning, network topology	-	Yes
Sen and Airiau, 2007	-	Learning	-	Yes
Savarimuthu et al., 2007 b,c	-	Leadership, network topology	-	Yes
Andrighetto et al., 2008, Campenni et al., 2008	Cognition	Imitation	-	Yes

Table 1. Mechanisms employed by simulation works in each phase of the norm lifecycle

A convention or (tacit) agreement betwixt us

Giulia Andrighetto, Luca Tummolini, Cristiano Castelfranchi, Rosaria Conte

Institute of Cognitive Sciences and Technologies
Via San Martino della Battaglia, 44
00185, Roma, Italy
{giulia.andrighetto; luca.tummolini; cristiano.castelfranchi; rosaria.conte}@istc.cnr.it

Abstract. The aim of this paper is to show that conventions are sources of tacit agreements. Such agreements are tacit in the sense that they are *implicated* by what the agents do (or forbear to do) though without that any communication between them be necessary. Conventions are sources of tacit agreements under two substantial assumptions: (1) that there is a salient interpretation, in some contexts, of every-one's silence as confirmatory of the others' expectations, and (2) that the agents share a value of not hostility. To characterize the normativity of agreements the Principle of Reliability is introduced.

1 Introduction

Conventions are social means for the sake of common ends. A common end needs not be a desire we pursue together (i.e. our joint desire to meet each other). A set of desires that are jointly co-realizable may suffice (i.e. our self-regarding desires to avoid collisions in traffic): coincidence of interests is, at least, agreement in desires[1]. Conventions describe a way to behave in recurrent situations, which is sufficient to obtain something we all want but which is at risk because of our reciprocal interference. Conventions are not necessary means though. They are arbitrary since some other way to behave might serve the same purpose. That is, our common interest (our ends in agreement) is to be fulfilled if our desires for the means are also in agreement, when at least another possible arrangement is foreseeable. To be useful, conventions should be stable: when established, conventions perpetuate themselves. And they are so because it is in the best interest of all of us to keep acting as we do, if the others do the same. Moreover this fact, as all the rest, is common knowledge between us, so much that, if one bothered enough to reason from the perspective of another fellow, it would discover that conformity to the convention is in the best interest of all the others and so be assured that the regularity will keep on.

[11] Two agents "agree in desires if exactly the same world would satisfy the desires of both; and a world that satisfies someone's desires is one wherein he has all the properties that he desires de se and wherein all the propositions hold the he desires de dicto. Agreement in desire makes for harmony" [20]. On the distinction between attitudes de dicto and attitudes de se see [18].

Dagstuhl Seminar Proceedings 09121
Normative Multi-Agent Systems
http://drops.dagstuhl.de/opus/volltexte/2009/1919

The idea that conventions are a peculiar kind of regularity in behaviour along these lines has been forcefully defended by David Lewis [15] [17], whose theory is considered by him as analogous to the one sketched by Hume in the *Treatise* while discussing the origin of justice and property.

According to this view, conventions *describe* a self-enforcing behavioural pattern; do they *prescribe* it too?

Many critics of Lewis' theory of conventions have been sceptical about his analysis, precisely because it seems that Lewis has missed the normative component. One way to put the critique being that conventions are not mere regularities but rules, not only regularities *de facto* but also regularities *de jure* [24] [8] [21]. Telling the truth when one is speaking in English is not only something that we *usually* do, it is something we *ought* to do. And the same is true for all the conventions we are parties of. Conformity to our conventions is not just what we happen to do, is something that is "required" from us.

Though often not acknowledged, Lewis' theory is able to readily accommodate these critiques. It is explicitly stated, in fact, that: "any convention is, by definition, a norm which there is some presumption that one ought to conform to (…) it is also by definition a socially enforced norm: one is expected to conform, and failure to conform tends to evoke unfavourable responses from others" [15].

What kind of norm any convention is, however, is not immediately clear.

Lewis suggests that there may be all sorts of reasons why, for any *particular* convention, one ought to conform to that particular regularity. If the convention originated by an exchange of promises, then one ought to act also to keep the promise; if the convention is also a social contract, then one ought to reciprocate the obtained benefit. Notwithstanding so, there are also *general* reasons why one ought to conform which are valid for any regularity that qualifies as a convention, for any population relative to which the convention exists, and for any situation the convention applies to.

Such general reasons derive from the fact that by conforming to a convention one acts in one's own best interest, and, at the same time, in a way that answers to others' preferences, *when they reasonably expect one to do so*. Both acting in one's own best interest and in the way that is in the interest of others (when they reasonably expect one to do so) are something that, according to Lewis, "we do presume, other things being equal, that one ought to do". If the former is a requirement of instrumental rationality, the latter stems from a moral principle that is, somehow, acknowledged by us. But is it so?

Alice has a good reason to expect Bob to do an action because John told her so. She completely trusts John; hence Alice has a reason to believe what John says. She really wants Bob to behave in that way and she reasonably expect him to behave so. Is this sufficient for Bob to be required to do the action in question? If Bob is not in any way responsible for what Alice believes, why ought he do that action?

Similarly, one can be reasonable in expecting conformity to a certain convention given widespread conformity in the population (e.g. it is reasonable to expect the next driver to keep the right given one's experience with what this population of drivers usually do) even without any direct experience with those of the others one is now dealing with (e.g. one's expectation about what the next driver will do is not grounded in one's experience with that driver). How is it, then, that such anonymous agent is responsible for expectations he has not induced? Though our intuition tells us that any anonymous driver ought to conform to the convention that prevails in that population, it is not evident why he is so bound since he bears no immediate responsibility for what anyone reasonably expects from him.

In order to clarify what kind of normativity characterizes any convention, in this paper we will argue that *conventions are sources of agreements*, though it is not necessarily by agreement that a convention is established. That a convention is an agreement is usually considered as a platitude, so much that once the notion of convention is understood, it is thereby clarified in which sense a behavioural regularity is also an agreement. Agreements however are not only agreements in desire that as a consequence produce regularities in behaviour. Agreements are specific kinds of *social relationships* between the agents, and are created with the aim to produce such agreements in desires (see Sections 4 and 6). Agreements are considered by Lewis as a means to produce a system of mutual expectations [15], but what is important for us, is that the converse also holds: a system of mutual expectations of the kind presupposed by a convention is a source of agreements. This suggestion, however, seems to be counter-intuitive given that conventions are typically maintained without the need of any communication between the parties. If this is true, how can agreements be established without communication? How can conventions be real agreements and not a way to behave *as if* we have agreed though we didn't?

It has been Hume's suggestion that a convention is an "agreement betwixt us, though without the interposition of a promise". The aim of this article is to clarify what kind of agreement is established, once a convention is in place. By doing this, the peculiar normativity of conventions will be also analysed. The normativity of conventions is the same normativity of agreements, because conventions become agreements, *tacit agreements* but agreements nonetheless.

2 From preferences to reasons to conform

Let's first rehearse what a convention is.

Few years after his first contribution on the topic [15], Lewis amended his original analysis, by offering the following definition [17]:

A regularity R, in action or in action and belief, is a *convention* in a population P if and only if, within P, the following six conditions hold:

1. Everyone conforms to R.
2. Everyone believes that the others conform to R.
3. This belief that the others conform to R gives everyone a good and decisive reason to conform to R himself.
4. Everyone who believes that at least almost everyone conforms to R will want the others, as well as himself, to conform.
5. R is not the only possible regularity meeting the last two conditions. There is at least one alternative R' such that the belief the others conformed to R' would give everyone a good and decisive reason to conform to R' likewise.

6. Finally, the various facts listed in conditions (1) to (5) are matters of common (or mutual) knowledge.

This definition is meant to capture the core of our common concept of convention whereby we are ready to acknowledge that a practiced regularity in acting or in acting and believing (condition 1), that everybody expects widespread conformity to (condition 2), that is arbitrary (condition 5) but serving our common ends (condition 4), and that perpetuate itself and it is stable because it is openly known that past conformity gives everyone a reason to go on conforming (condition 3 and 6), is what we would indeed consider one of our conventions[2].

Lewis has amended his 1969 analysis in several ways, but one change was particularly relevant to his original target, that is, the explanation of what convention underlies the use of a certain language by a population. Since clause (3) was originally formulated in terms of a conditional *preference* for conformity, the only acceptable regularities were in action alone: it makes no sense to prefer to believe something, since you cannot choose what to believe. As a consequence the convention governing the use of a language was characterized as a *convention of truthfulness* in that language, whereby only speakers conform to the convention, and, by doing so, coordinate with past speakers who truthfully used that language in the past [15]. Differently, the amended definition makes room for a regularity in *action and belief* to count as a convention since others' conformity provides one with a *reason* either to do or to believe something. The formulation in terms of reasons for conformity (instead of preferences for conformity) opens the way for coordination between speakers and hearers so that the convention whereby a population uses a language becomes a *convention of truthfulness and trust*, that is, a regularity in which conformity for speakers means to do something (i.e. speak truthfully) and for hearers to believe what speakers say since both share an interest in communicating and each other conformity is a practical or an epistemic reason to conform.

3 Generalizing Lewis: trust by convention

Once, however, convention is defined in this way it is also clear that trust, properly defined, is not peculiar of conventions of language alone.

Trust, in fact, is both a state of mind and a behaviour [3] in which an agent expects and wants that another agent does something, relies on this agent to behave in this way, and does in fact delegate the fulfilment of one's own desire to another agent. By trusting another agent, one makes oneself vulnerable; one exposes oneself to the risk that the other will not behave in the expected way and so frustrating one's desires.

Crucial for trusting is *reliance* on an agent for something, and not just reliance on something happening [13]. When we rely on something happening, say that the train will arrive on time, we assume that it will happen (usually because we believe that it

[2] Many of course have challenged this analysis under several different aspects. Here we will just assume it as correct, and focus on how, within such a framework, the normativity of conventions can be accounted for. For a critical assessment of Lewis' theory see [8]. For a recent account see [10].

will happen) and plan or intend accordingly. Differently, when we trust an agent we rely on him *as* an agent, that is, as an autonomous entity driven by his own beliefs and desires that are his reasons either to believe or do something. That is, when we rely *on an agent* to behave in some way, we assume that such agent will behave in that way and we plan or intend on this basis *because the agent's behaviour is based on his reasons*, not just because he his coerced to behave in that way. If I coerce you into giving me your pocket, I rely on the fact that you will give me your pocket but I do not rely *on you* to give it to me; there's no question of trust in coercive interactions. By the same token, trust also presupposes that the trustee is not motivated by a hostile attitude towards the trustor, so much that the trustor at least believes or assume such non-hostility in those the trustor rely on [3]. Trust is a fundamental non-hostile attitude[3].

Trust is relative to a desire one is pursuing and whose fulfilment depends on another agent's behaviour[4]. Desires can be either epistemic desires (i.e. the desire to know something or to know whether something is true or not) or practical ones (i.e. the desire that the world be in some way). Correspondently, reliance on somebody to behave in some way can be either for an epistemic or practical desire. That is, if I epistemically rely on you, I rely on you to do something in order to fulfil an epistemic desire of mine, something that typically happens by way of communication[5]. In such a case, epistemic reliance entails that I assume that you will truthfully communicate with me because you are motivated (for some reason) to so act, and, on this basis, believing what you want me to believe. This is the kind of trust that Lewis had in mind, where trust is coming to believe something. On the other hand, when I practically rely on you, I rely on you to do something in order to realize a state of affairs that I desire. If I rely on you to drive on the right side of the road, such practical reliance entails that I assume that you will so drive since you have a reason to do such an action, and I will behave accordingly. In both situations, by coming to believe something or by acting on the basis of my expectation about you, I trust you.

Finally, one trusts on the basis of reasons. But what are the reasons to trust another? Sometimes trusting may be 'irrational', as when, by making oneself vulnerable, one thereby creates a selfish reason for another to exploit such vulnerability[6]. Other times, trust is perfectly reasonable as when one relies on another to do something simply because it is also in the interest of the other agent to act in that way. Even if in this case trust is reasonable and more secure, it is not of course without risks given that the other could simply change his mind and act differently.

What is then the relation between trust and conventions?

According to the definition of convention given above, in any convention, the agents do conform to some regularity, want the others to conform, expect future conformity of their fellows, and this belief is a reason to conform (i.e. a practical reason to do an action or an epistemic reason to believe something). Given this, it is clear that *any act of conformity to a convention is also an act of reliance on the others to*

[3] See Section 6 for the relevance of not being motivated by hostile attitudes.

[4] That is, the trusting agent believes to be dependent on another one to obtain something he desires [3],[4].

[5] Though this is not necessary, see Section 8.

[6] If one extends a loan to another assuming that the other will do his best to repay it, one also gives the other a selfish reason not to repay it; see [1].

conform: the reason to conform is also the reason to trust, to rely upon the others and doing something accordingly. Since, however, by conforming one trusts in others' conformity, that is, in their trust in oneself, *the regularities that count as conventions are regularities of reciprocal trust*. Moreover, since the expectation of conformity is a reason to conform, trust is based on trust: I have a reason to trust you if you trust me and you have a reason to trust me if I trust you.

More precisely: a regularity R in reciprocal trust in a population P is a convention if and only if the following six conditions hold:

1. Everyone conforms to R, that is, everyone reciprocally trusts each other.
2. Everyone believes that others conform to R, that is, everyone believes that the others trust in oneself.
3. This belief that everyone conforms to R (i.e. rely on each other) gives everyone a good and decisive reason to conform to R himself (i.e. to rely on the others). That is, the belief that everyone reciprocally relies upon each other is a reason for everyone to rely on the others. This reason can be for practical reliance, if conforming to R is a matter of reliance on the others to act in a certain way and acting oneself accordingly. This reason can be for epistemic reliance, if conforming to R is a matter of reliance on the others to act in a certain way and believing oneself accordingly. In the case of a regularity of practical reliance, some desired end may be reached by relying upon the others and acting accordingly, provided that the others also rely upon on each other; therefore he wants to rely on the others and act, if they so rely and act. In the case of a regularity of epistemic reliance, his beliefs together with the belief that the others practically rely upon himself are premises that deductively imply or inductively support a conclusion, and by believing this conclusion he would thereby conform to R (i.e. he would epistemically rely on the others).
4. Everyone who believes that the others conform to R (reciprocally trust each other) will want the others, as well as himself, to conform (i.e. to trust on oneself).
5. R is not the only regularity meeting the last two conditions. There is at least one alternative regularity R' in reciprocal reliance which would perpetuate itself instead of R.
6. The various facts listed above in conditions (1) to (5) are matters of common knowledge.

Any convention then is always a form of reciprocal trust, which is sustained by past reciprocal trust, and that breeds future trust. Such reciprocal trust is reasonable since by trusting we are able to agree in the choice of the means to fulfil each of our individual end. Reciprocal trust can originate in several different ways, for example by explicit agreement. However, a regularity of reciprocal trust qualifies as convention by the way it perpetuates itself, and not by the way it originates. There is *trust by convention* whenever it is our reciprocal trust that, together with our desires for the end, gives us a reason to keep on trusting.

Generalizing the definition in this way is faithful to Lewis' analysis because no modification or additional clause has been proposed. Whether it is also fruitful to understand the peculiar normativity of conventions will be explored in what follows.

Since we intend to argue that such normativity stems from agreements, in the next section we turn to this issue.

4 Agreements without promises

It is natural, and correct, to view the practice of promising as a social device for making agreements. It is also natural, but wrong, to consider agreements primarily as ‘an exchange of conditional promises’[7]. Though it’s true that such an exchange creates a binding agreement, even my unconditional promise to you is sufficient for creating an agreement between us on something I will do, no matter what. Mutual conditional promises may be the natural model for contracts, but they hardly are the general analysis of agreements, at least if, as Hume has suggested, agreements might exist between us without the interposition of any promise.

In the contract view, agreements create obligations (and rights) on the parties entering into it due to such exchange of conditional promises. However it is sensible to consider promises just as one possible way to create agreements. Another opportunity is to avail oneself of a suggestion of a third party that, if it meets the interests of all, might be jointly accepted. However, that an agreement be mutual is also dispensable. Giving permission, for instance, is a way to enter in an agreement, originating only unilateral obligations and rights. When an agent gives the permission to another to do something that he has the power to prevent, there is an agreement between the two that enables the latter agent to do some action. In this kind of agreement, an agent becomes obliged not to interfere with the other one, who at the same time acquires the right to act as agreed upon. But given that no promise has been formulated, where does exactly such normative consequences come from?

An answer to this question is postponed to the next two sections because it is useful, first of all, to clarify what an agreement *primarily* is.

When there is an agreement between some agents, say Alice and Bob, the *consent* of at least one of them is necessary. When one consents, one is consenting *somebody to something*. Hence consenting creates a social relation between at least two agents.

But what one is consenting to? Sometimes Bob consents Alice to do something, like when he consents her to use his car. Other times, Bob consents to do something himself, like when he accepts to pick the children from school. Other times it happens both that Bob consents Alice to use his car and that he gives her the keys. In all these situations, by consenting an agreement is established. In any case, consenting is related to the fulfilment of another agent’s desire *that one can interfere with*. This desire might be to do something that one may impede (negative interference). Other times, the desire is that one does something to favour another one (positive interference). Or a combination of both at the same time, like when Alice’s desire to do something depends on Bob creating some favourable conditions, something which she also desires. What is common to all these cases, it that an agent has the power to interfere somehow with another agent’s desire, and when the former consents that the latter fulfils such desire, it is entailed that the former does not interfere negatively

[7] [26], see also [15]; [9] for a critique of this view.

with it or that he interferes positively with it. For simplicity, from here on, we will just mention the negative interference situation.

An agreement then creates a social relationship between the parties, and presupposes a pre-existing *asymmetrical* social relation of dependence between them. When there is an agreement at least one agent that could (has the power to) interfere, is not interfering.

However something stronger is needed to have a real agreement. Though it is true that a lion that is not hungry is consenting a gazelle to wander around him safely, the gazelle does not have the lion's consent to do so and there is no agreement between them to this purpose. The gazelle does better to be ready to run as soon as the lion manifests any change of mind; she may exploit such temporary loss of interest but not rely on the lion only because the lion does not have a desire to interfere with her. Differently, her reliance would be more justified if the lion could be able to communicate his decision (i.e. intention) not to interfere with the gazelle, that is, to express his *consent*. Hence, one's consent (not just a behaviour that happens to consent) to the fulfilment of a desire of another agent is there when *one has the intention not to interfere with such desire fulfilment*. To be able to formulate such an intention one obviously is to be able and in condition to interfere, thus this condition presupposes the truth of the former.

This unilateral consent, however, is still not enough. Suppose Alice and Bob live together, and Bob has bought a car. Though the car is legally owned by him, between them, Alice may not 'acknowledge' it as Bob's because she does not consider the matter of who uses it as entirely up to him; she does not consider this choice as depending on him alone. She knows that Bob has the keys, and that he has some sort of legal power to interfere with her free use of it (she could be charged of theft, for instance). Alice also knows that she has Bob's consent to using the car whenever she wanted to, but still she contests this power over her. In this case, though all the conditions above might be true (i.e. Alice objectively depends on Bob, and Bob has decided not to interfere with Alice), there is still no agreement between them. It might be said, that Alice uses the car despite the fact that Bob can (he has the power to) interfere with her. To have a full agreement, then, there must be also *an acknowledgement of the power of interference of the agent who, in fact, has such power*.

Suppose now this variation of the example. Though the car is legally Bob's, it is Bob that reject his own power over Alice, as far the use of the car is concerned. Alice may consider the matter of who uses the car as up to Bob, but Bob himself contests this fact. If Alice asks his permission to take the car, Bob replies that she does not have to ask for it, that it is her choice whether to take it or not. In this situation, again, there is no agreement between them that Alice uses the car, because, between them, though Bob has the power, he does not 'value' it.

What is, then, to value a power, and what relation does it bear with power acknowledgement?

Valuing one's power is not simply desiring to exercise it because it may happen that Bob can indeed desire to use it against Alice in a moment of sudden anger but then hates himself for such desire given that Bob in fact contests such an asymmetrical relation between them. Bob does not desire to desire in this way towards Alice, that is, he does not value his power over her. One values one's power when one *desires that the use of one's power is motivated by one's desire to do so*, that is, when one desires that it depends on one's choice whether to interfere or not. Hence, in the latter example, despite the fact that Bob can in fact interfere with Alice and he could come to desire to exercise his power over her, he does not want to be so motivated, at least when it comes to have a power over her. This example makes clear that, to have an agreement, *the agent values the power he has over the other one*, the agent values the fact the he is able and in condition to interfere with the other[8].

Power acknowledgement, differently, is the *acceptance* of such power, that is, the decision to forbear to resist to the exercise of the power over oneself if the other wants to exercise it. Acknowledging the power of another makes manifest one's fundamental non-hostility towards the other: to be prepared not to pursue something if this happens to be against the other's desire. While in giving one's consent, one accepts something one can interfere with (i.e. intends not to use a power of interference one has), power acknowledgement, differently, is just the acceptance of the use of such power, that is, the intention not to resist to the decision of the other agent. Both *valuing one's power of interference* and *the acknowledgement of another agent's power of interference* are necessary conditions to enter in an agreement.

Consider now this example. Alice wants to use Bob's car tomorrow, she has his consent, and she acknowledges his power on this matter. Notwithstanding so, to be safe in case something happens, Alice books a taxi for tomorrow morning. Suppose that Alice is quite sure that, intending not to interfere with her use of the car, he will so behave. But still she is worried that something unexpected might turn out. Assuming a worst-case scenario (which she considers highly improbable anyway), Alice decides not to rely on Bob. It seems that, in this situation, if Alice does not *uptake* Bob's consent, no agreement between them has been established. And *uptake* precisely *is such reliance on one's part on another agent's intention not to interfere with one's desire fulfilment.*

Finally, even if such condition is needed, it is not in itself sufficient for creating an agreement. In fact, Bob may know that she has a very important meeting tomorrow and that she needs the car. To avoid creating any obstacle, Bob decides to refrain from taking the car in the morning but he does this without that she realizes this intention of not interference, hence she does not uptake his consent though she would in case she knew about it. While Bob's intention of not interference is present, her ignorance of such intention makes it the case that they have no agreement that Alice uses the car today, and she may decide to call the taxi. Knowing that another agent has the power of interfering with oneself, and knowing that the other intends not to exercise such power is needed to have an agreement. But, as it is standard in many social interactions, even such first-order knowledge isn't enough to have an agreement because Alice may know this fact while Bob does not know that she knows it and, on this basis, Bob may think she will act otherwise and so in the end deciding to pick the car on the assumption that Alice may have decided to call a taxi, and so on for all the levels. In any agreement, then, an epistemic condition is necessary, that is, there should be *common knowledge* of the intention to not interfere. The same reasoning supports also other epistemic conditions. An agreement, in fact, cannot be in place unless the agent, who is consenting to the other's desire fulfilment, knows about such desire in the first place. And again this fact must be out in the open by being common

[8] On valuing and second-order desires see [7] and [20]; for a critique see [31].

knowledge that an agent has the power to interfere with a desire of another one who in fact has such desire. The acknowledgment of such power made by the other agent also must be matter of common knowledge, given that an agreement basically is a way to obtain something one wants without coercing the other to do so. And, finally, both the valuing of one's power and that the uptake of the consent are again matter of common knowledge between the agents.

Let's take stock.

A social relationship between at least two agents is an *agreement* between them if and only if the following five conditions hold:

1. The agent having the power of interference intends (for some reason) not to interfere with the other agent's desire fulfilment (consent condition);
2. The agent having the power of interference values his own power (valuing one's power condition)
3. The agent, who is subject to interference, acknowledges the power of the other one, that is, he intends to refrain from pursuing his desire if the other desire that he so behaves (no coercion condition);
4. The agent, who is subject to interference, relies on the consent of the other one, that is, intends to pursue his desire on the assumption that the other one intends not to interfere (uptake condition);
5. All conditions above are common knowledge.

An agreement of this sort may be called unconditional, in the sense that one does not give one's consent on condition of another agent's consent. Differently, an exchange of conditional promises gives rise to a conditional agreement in which each consent is conditioned on the other. Contracts, for instance, are instances of conditional agreements.

Moreover, on this analysis, it is also evident that there can be agreements without promises. Agreements are particular kinds of social relations between the agents, and a promise is one possible way to establish such relations (see also Section 6). Other possibilities, such as a mere exchange of a request and an acceptance or a mere unilateral permission without any request, make it clear that no promise is indeed necessary.

5 The principle of reliability

All agreements have normative consequences, even those that are unconditional and established without the interposition of a promise. However, on the present analysis, an agreement is primarily a social relation characterized by specific motivational and epistemic conditions that are true of the agents entering into it, and so no normative relation has been so far mentioned. How, on this account, is it possible to explain the 'obligation' of the consenting agent, and the corresponding 'right' to do or to obtain what an agent has been consented to? Or, differently put, what is the wrong of infringing an agreement?

In our view, the wrong of violating an agreement not made through promises is of the same family of the wrong one would commit if the agreement were promise-based. Both situations, in fact, pertain to a more general kind of social interactions that are wrong in relation to “what we owe to each other when we have led them to form expectations about our future conduct” [27].

The moral Principle of Fidelity put forward by Thomas Scanlon in this seminal paper was intended to account for the wrong of breaking a promise, and, as such, may be too strong for the kinds of agreement without promises we are after. However Scanlon has also insisted on several moral principles bearing family resemblances with each other given that all are related to the elicitation of expectations in others. To account for the normativity of agreements without promises the so-called principle of Loss Prevention could be enough [27]. This principle requires that *one that has intentionally or negligently led someone to expect that one will follow a certain course of action, and has reason to believe that that person will suffer significant loss as a result of this expectation if one does not fulfil it, must take reasonable steps to prevent that loss, that is, he ought to warn, fulfil the expectation or compensate*.

The fact that the principle is not just to prevent another agent’s desires frustration but losses, indicates that some form of reliance is presupposed for the principle to be applicable. Suppose, in fact, that Bob had, somehow, led Alice to expect that he won’t take the car tomorrow morning, say because he knows that she heard him accepting a lift from a colleague on the phone. Bob knows that she cares about this fact given that she needs the car. Still Alice decides not to rely on Bob as for having the car at her disposal tomorrow, and, to be completely safe, she books a taxi. Knowing this, Bob is under no obligation towards Alice, not even to warn her that in the end he will take the car. Though taking the car might be something she desires more than just taking whatever means of transportation, the frustration of this desire of her is not a loss Alice incurs with, it is not something she has and she wants which she is deprived of, hence the principle of Loss Prevention does not apply. Under this respect, even in the case that she had relied upon Bob and decided not to call the taxi, the very fact that the desire is frustrated when Bob instead took the car is not a real loss[9]. However in relying on Bob, Alice has in fact lost something she had before her. She has paid some costs, opportunity costs as the economists call them, which are the available alternatives of actions she had and which she has renounced to pursue by counting upon Bob’s car being available.

At least for the aim of this article, then, the way Judith Thomson has defended a similar principle seems better suited [29]. Thomson, in fact, argues for the validity of a Word-Giving Thesis in which, when an agent invites another one to rely on the truth of a certain proposition, which invitation the latter agent accepts (or uptakes), then the latter agent acquires a claim (i.e. a right) against the former one to its being true. This way of formulating the moral principle bears two main advantages over Scanlon’s: firstly, it makes explicit the relevance of reliance or uptake in the process, and, secondly, it generalize it towards whatever proposition one may rely upon beside those that refer to an action one will do in the future.

[9] Though it can be so when I consider the desire to have the car not as something I am to achieve but as something already achieved and to be protected, see for this possibility and its psychological plausibility [22].

However, though one can induce reliance, one can *allow* reliance as well and in such a way to have normative consequences.

Consider again the example above: Alice heard Bob's conversation with somebody else and, as a consequence, she comes to believe that Bob will not take the car and she relies on it. In this case, Bob has unintentionally induced in Alice some kind of reliance. We have suggested that by acting on these expectations about Bob, she will incur in some losses, and so the principle of Loss Prevention might apply. But is it so? After all, such induced reliance in this case is not intentional; can Bob be responsible for Alice's unilateral decision to rely upon him in this situation? It seems correct to say that though her reliance has been only involuntary induced, at least Bob has *allowed* her to rely on him. More precisely, in fact, *to allow a belief or an action is to have the power to disconfirm another's belief (which is a reason to believe something else or to act in some way) and to forbear to disconfirm it*. If hearing what Bob has said on the phone is a reason for Alice to believe that he will not take the car tomorrow, then this belief is obviously something that Bob can disconfirm. By not disconfirming such belief, Bob is also allowing her to believe in this way. Granted this, as such this form of allowing is still not sufficient for an agent to acquire a claim against another one. Suppose in fact that, immediately after having realized her reliance, Bob tells her that what is true is just that he does not confirm that he will take the car (which is the same of not disconfirming the belief that Bob will not take it) and nothing more than that. Can Alice hold him responsible for her losses if in the end he decides to take the car despite her unilateral reliance? It seems not. Suppose differently that just after his conversation on the phone and knowing that she needs the car, Bob turns to Alice and say 'yes, you heard correctly. I won't take it!'. By confirming a belief that he has unintentionally induced in her, Bob then become obliged towards her to warn in case he changed your mind, or, if it's too late, to do as expected or to compensate. Because such confirmation of the belief logically entails the absence of a disconfirmation, even in this case Bob has allowed her to believe something, though not passively (i.e. by forbearing to disconfirm it) but actively (i.e. by confirming it). It is this form of 'active' allowing that is necessary for the moral principle to apply when one does not induce intentionally reliance in others[10].

Finally, there are also cases in which one actively allows other agents' reliance on oneself *that one has not in any way induced*.

Suppose for example that Alice believes that Bob will not take the car tomorrow because John told her so, and that she relies on him for having the car. Bob knows about all this and he allows her to believe it (i.e. he forbears to disconfirm such belief). If she just act on this basis, and she does not know that Bob knows about her reliance, it seems that at most Bob should warn her if the belief is false, but if this is so, it is just out of sheer altruism[11]. If, differently, Bob has confirmed this expectation of

[10] Scanlon's principle of Loss Prevention indeed mentions also leading expectations negligently, besides doing it intentionally. However, negligence implies having not paid *due* care to avoid such reliance, and so it cannot be evoked to explain, without circularity, a principle which normatively demands such behaviour. Differently, our notion of active or confirmatory allowing has not such problem.

[11] The reason why common knowledge of another's forbearance to disconfirm one's beliefs may change the situation will be discussed in the Section 7.

her, for instance by nodding, Bob has actively allowed her to rely on him, and, from there on, he is responsible for her possible losses even if he has not induced that belief in the first place. Again, when it's too late for warning, Bob ought to fulfil the expectation or compensate.

To sum up, according to the view adopted in this paper, an agreement has normative consequences because the agent consenting another one to fulfil his desire is *either intentionally inducing or actively allowing uptake* on the consent is concerned (i.e. reliance that the former one intends not to interfere with such desire fulfilment), and, by doing so, undertakes a *duty of reliability* against the other one and creates a corresponding *right to rely*. Reliability is normatively required to prevent losses caused by intentionally inducing or actively allowing such reliance. One way in which such principle can be explicitly formulated is the following: *if one intentionally induces or actively allows another agent to rely on the truth of a certain proposition, then the latter one acquires a right to reliability (i.e. to be warned if the proposition turns to be false, or, in case the proposition is about the future action of the former one and it is too late for warning, a right that the former one acts so as to make the proposition true or to be compensated for the incurred losses)*. For these reasons, we name such a principle: the *principle of Reliability*.

6 The normativity of agreements and the value of not having hostile attitudes

Thus, by establishing an agreement between the agents at least one of them intentionally induces or actively allows the uptake of the other one. From this it follows that the uptaking agent acquires a right to rely on the other one. But what exactly he has a right to rely on?

We have suggested above that if there is an agreement between Alice and Bob that Alice will take his car tomorrow, his taking the car is doing something wrong. Under this perspective, by uptaking the consent, one acquires a right on a certain behaviour: i.e. that the other does not interfere with his desire fulfilment. But it seems even more than this: even if Bob does not take the car but afterwards he manifests some uneasiness because she has taken it, it seems again that Bob is doing something wrong. If agreements were there only to rule behaviours, what Bob has done should be enough for complying with its terms. However it looks like that it is not.

To understand why it is so, and what the peculiar normativity of an agreement is, consider the difference between the mere fact that some agents agree in their desires and the fact that there is a social relation of agreement between them.

When they agree in desires "the same world would satisfies the desires of both" [20] possibly without any social relation between them whatsoever. Differently, when there is an agreement there is also a social relation between them that aims precisely to create such agreement in desires *but facing the fact the things could have been different*. In fact, as we have noted above, an agreement presupposes an asymmetrical relation of power and dependence between the agents so that one can influence the fulfilment of the desires of the other.

However by acknowledging such power, one also signals one's basic *non-hostility*, that is, one's desire not to be motivated to do an action against the desire of an agent that has and values his power of interference. Correspondently, for whatever reason an agent decides to do so, by giving the consent, an agent also signals that the fulfilment of such desire 'agrees' with his own desire in the present conditions (i.e. the agent for some reason desires not interfere with the other). Thus, *an agreement results in the fact that the agents mutually know that their actual desires agree*: they are jointly co-realizable and they are so without any coercion.

Consider now the principle of Reliability. The similar principle of Loss Prevention is justified for Scanlon on a contractualist basis by the fact that "it is not unreasonable to refuse to grant others the freedom to ignore the losses caused by the expectation they intentionally or negligently lead others to form" [27]. One reason to refuse such freedom is readily available if *the agents share a value of not being motivated by hostile attitudes*[12]. In fact, ignoring such losses, when one has intentionally induced or actively confirmed an expectation on oneself, would be tantamount to be motivated by a hostile attitude: either one desires that the other incurs in those losses or at least lacks the desire that the other does not incur into them. Let's assume, then, that our agents share this value so that the principle of Reliability, as it has been here formulated, would just follow.

According to Lewis' dispositional theory [20], this would be a value *de se*, that is, *a property that the agents are disposed to desire to desire (i.e. to value) under ideal conditions*. The value of not being motivated by a hostile attitude amounts, then, to the fact that, if the agents are under ideal conditions, they are disposed to desire to desire to have such a property. Moreover, given that being motivated by a hostile attitude is being motivated to frustrate the desires of another agent, the compliance with such value requires them to revise their possible first-order hostile desires in a way that would inevitably result in the creation of harmony in the population, that is, in desires that agree.

Sharing the value however does not necessarily mean that the agents *will* behave according to what is required of them in the present conditions. It would of course in case they were in ideal ones, but no one is a saint, that is, no one lives always up to one's values. However, between agents that enter in a social interaction, such value can at least ground the *presupposition* that the other fellows will be so motivated, otherwise the best one can do is to avoid any possible contact with them.

What is then, on this basis, the peculiar normativity of agreements as social relations?

Recall that in giving one's consent, one induces or actively allows another agent's reliance on the consent, that is, not just on the observable behaviour of not interference but, more specifically, on the decision not to interfere. Moreover, given the details of the social interaction between the agents, it is also manifest that the decision is based on the fact that the desire of the consenting agent agrees with that of the other. It is on the decision based on this 'agreeing' desire that the other rely upon, if he wants to be non-hostile with the other. As a consequence, and given the principle of Reliability, those who uptake a consent acquire a *right to such decision of not inter-*

[12] Another reason would be available to them if they shared a value of assurance as Scanlon suggests [27].

ference based on an agreement in desires. Thus, the consenting agent that is willing to enter an agreement is not only obligated not to interfere (i.e. not to take the car himself), he is in fact bound *not to change his mind* otherwise the basic non hostile attitude of the agent would be frustrated: he is bound not to come to desire to interfere with the other. *When giving one's consent, one is obliged to keep one's desire in agreement with the other*. For this reason, it turns out that it is illegitimate even the expression of Bob's uneasiness with what Alice has done since such reaction on his part would signal that Bob has indeed changed his mind.

Is the other also bound similarly? We think so. In fact, the consent is given, and the decision is taken, on the assumption that the other agent has the desire in question (i.e. she wants to take the car): Bob relies on this fact and Alice has induced him to so rely. Hence Alice is bound too not to change her desire, on pain of being hostile with Bob, given that the opportunity costs he has paid to eventually decide not to interfere with her would then become just induced losses.

Hence, even in an unconditional agreement as this one, there are reciprocal obligations and reciprocal rights. *By establishing an agreement between them, the agents become reciprocally obligated and entitled to keep their mutually known desires in agreement*[13].

Does this entail hence that an exchange of promises has indeed occurred? No. By promising one creates the expectation that the promisor will do an action in the future *unless the other consent to not doing so* [27]. When giving one's consent without the interposition of a promise a timely warning can still be enough to release oneself from an obligation, at least when the other has not lost valuable alternatives to satisfy his desire. Agreements not based on promises are just weaker than agreements based on promises. They aim to create and protect desires that agree, and they do so for agents that share the value of not having hostile attitudes.

7 The ambiguity of silence and tacit confirmation

Now, suppose that it is common knowledge between Alice and Bob that Alice wants his car tomorrow morning, and that Alice believes that he will not take it because tomorrow is Monday, and on Mondays Bob never takes it (maybe just because it is his habit to act in this way or because the traffic on Monday mornings is more intense than in the other days and Bob hates to be stuck in traffic). Given that she believes that he has his own reasons for not taking the car, and she knows that he usually act in this way on Mondays, it is reasonable for her to expect Bob to behave in this way this Monday too (i.e. she believes with some probability that this will happen). Alice so believes this that she relies on him for not taking the car, and she decides to go to the meeting with his car. All above being common knowledge between them, she also observes that Bob has kept silent about the truth of this belief until Monday morning. However, just when the time has come, Bob decides to take the car, say because it

[13] More precisely, the obligation is to keep one's *first-order* desires in agreement. Such an obligation can be seen as a reason for all the parties to the agreement to have a second-order desire that their first-order desires keep motivating their behaviour. Those second-order desires would motivate the agents to do whatever they can to avoid to revise their first-order desires.

happens that today he needs the car for some unanticipated errands. Has Bob done something wrong?

It is foreseeable that having incurred in some losses, Alice may resent Bob's late decision, and she may even protest about such sudden change of mind. But, is she entitled to anything? Is Bob under a sort of duty towards her? In case she thought that way, Bob could legitimately claim to have not given her any consent to use the car, not even acted in order to make her believe something about him, that is, not even implicitly consenting her to something. So why would be Bob responsible for her losses? In the end, he has not intentionally induced any reliance on himself nor he said 'yes' or any other kind of confirmation because, by assumption, no communication between them has occurred.

Granted this, however something strange has indeed happened.

The closer she come to the fulfilment of her expectation, *the more she feel sure about such fulfilment and entitled towards the other acting as expected*. It is a fact that, though Bob knew about her belief, he kept silent until the moment has come, that is, that Bob has not disconfirmed her belief.

Suppose that Alice has interpreted this silent behaviour as a *confirmation* of her belief that Bob won't take the car, and then she has felt that such confirmation has somehow entitled her to have the car. But what kind of confirmation is this given that they do not communicate? Is it reasonable to read the other's silent behaviour in this way? And how can the omission of a disconfirmation create duties and rights?

To understand this issue more clearly, suppose that Alice is a Bayesian rational agent, that is, suppose that H_i is her hypothesis that Bob will not take the car tomorrow that is characterized by a subjective probability $p(H_i)$, representing her degree in belief in H_i. Because beliefs are represented by a well-defined additive probability function [27], her degree of *dis*belief in H_i is given by 1- $p(H_i)$. We can imagine such beliefs be warranted by inductive reasoning in which Alice has acknowledged that there is a pattern governing Bob's behaviour such that, almost on every Monday Bob does not take his car or, simply, that not taking the car on Monday is his best choice given his desire not be stuck in traffic.

Suppose that given Alice's concern on what Bob will do this Monday, she starts looking for additional evidences for her belief that he won't indeed take the car. Assuming, as we have done above, that everything is common knowledge between them, she happens to notice that Bob keeps silent about the truth of this belief she has about him, though he knows that she has decided to rely upon him.

The observation of silence, from a Bayesian perspective, can be treated as a 'datum' S for determining whether Alice's belief about Bob is true or false. Hence, by applying the Bayes' theorem, the belief can be updated accordingly. Moreover, such update of H_i must be determined relative to its complement $\neg H_i$, as the usual formula makes clear:

$$\frac{p(H \mid S)}{p(\neg H \mid S)} = \frac{p(S \mid H)}{p(S \mid \neg H)} \bullet \frac{p(H)}{p(\neg H)}$$

As a Bayesian rational agent, Alice is interested in the impact of the fact that Bob is silent on her belief that Bob will not take the car tomorrow, which amounts to calculating the probability that her belief is true, given that she has observed his silence. To do this, as a Bayesian rational agent, she needs to compute the *posterior* (i.e. the odds that H_i is true in light of what is known after the observation of S) that equals the *likelihood ratio* (i.e. the second term from the right representing the information value of S with respect to the truth of H_i) multiplied for the *priors* that H_i and $\neg H_i$ are true before the observation of S. In such an inference, in case the probability of observing S when H_i is true differed from when is not true, the likelihood ratio would be different from 1, and the posterior would also differ. In particular, the datum (i.e. Bob's silence) favours the hypothesis H_i when the posterior odds are greater, and this happens when the conditional probability of his silence given that Alice's belief about him is true is larger than the conditional probability of Bob's silence given that her belief about Bob is false. In such a case, it is said that the observation of S is *diagnostic* of or *confirms* H_i and not $\neg H_i$.

Silence clearly is ambivalent evidence in that there are both reasons for believing that it supports Alice's belief about Bob (if Bob does not want to take the car, he does not inform Alice that he will instead take it) as well reasons to believe that it can *dis*-confirm my belief: it may be possible that Bob could not reach her in time or that he has forgotten her desire to have the car, or that he simply does not care about Alice enough to let her know something relevant for her, or that Bob wants to harm her on purpose and so on. Whether the evidence is relatively more confirmatory than not is a contingent matter, and depends on the ratio between the *known* conditional probabilities of observing silence on condition that my belief is true or false. If she is Bayesian rational agent, she compares these information values before updating her belief.

There are, however, (psychological) reasons to believe that Alice, as all of us, is not so rational.

It is in fact one of the "best known and most widely accepted notion of inferential error" [6] that human reasoning gives undue weight to evidence that supports one's beliefs while discounting evidence that would tell against it, and this tendency is called *confirmatory bias*[14]. A confirmatory bias can be discovered in many different situations in which one assesses the truthfulness of one's beliefs. However the scientific evidence is particularly vivid when one is both *concerned* in what one believes (the so-called motivated confirmation bias) and the evidence one is evaluating is *ambiguous* (i.e. it is partly supportive and partly not *without exactly knowing how much it is so*). In this kind of situations, there is a very strong tendency to interpret information in ways that are partial to one's beliefs, and in particular, in ways in which the positive side of the evidence is overemphasized.

On the basis of these empirical facts, it is seems plausible to assume that there is an analogously strong tendency *to read other's silence*, in the kind of situations we are interested in, *as a positive evidence for one's belief*. In fact, the ambivalence of silence would not be too much of a problem if silence were not often an *ambiguous* evidence in that one is not so sure on how to assess such ambivalence, whether the positive support to one's hypothesis is more likely than the negative one (Ellsberg 1961). In the case at hand, ambiguity about the evidential value of silence can be seen as a form of uncertainty about the relative conditional probabilities of $p(S|H_i)$ and $p(S|\neg H_i)$. The agent does not know what the likelihood ratio is because it is as if he considered as reasonable, in the present circumstances, more than one distribution of

[14] See [23] for a review of the relevant psychological literature; see [25] for a mathematical model, though focussed on a different aspect of the confirmatory bias.

conditional probabilities of observing silence, given that the hypothesis is true or false.

If we accept the confirmatory bias, it may be suggested that, in contexts where we already entertain the relevant belief, we update it by adopting the *best* expectation that could be associated with the observed evidence, which is the one that would *confirm* the belief already accepted. In other words, silence regarding one's belief, that is, the forbearance to disconfirm such belief means, for the agent holding the relevant belief, that the other one will act *as expected*.

To interpret silence in this way, one must think that the other is not hostile towards oneself, otherwise, if this were not the case, if he believed in the other's hostility, then the negative side of evidence would be maximally relevant. However, as we have assumed above, such non-hostility is a reasonable presupposition for agents that interact with each other. Under this presupposition of non-hostility, it is reasonable to consider that the 'natural' meaning of silence is confirmatory.

It is then understandable why the more Alice is close to fulfil her desire that Bob does not take the car, the more she is *sure* that he will not take it. Supposing that she has checked upon him several times until Monday morning arrives, each time Bob's silence has confirmed her belief possibly up to certainty.

So far so good for the expectation that Bob will not take the car becoming firmer (i.e. confirmed). But what about the fact that she also feels *entitled* that he does not take it?

First of all, given that the confirmatory meaning of silence is salient between them (Bob is a confirmatory agent just as Alice is) and he knows that he has not disconfirmed a belief she had about him, Alice has reasons to believe that Bob cannot but assent to her interpretation (at least from the perspective of bounded rationality): that Bob's silence means that Bob will act as expected is 'natural' or *salient* in this context (i.e. it is the obvious interpretation for confirmatory and non-hostile agents). If Bob has reasons to *assent* to Alice's belief, he has reasons to believe that it is reasonable to believe something in those circumstances and so he has reason to believe that he has as a matter of fact confirmed Alice's belief about him. If the salience of precedence suffices to justify the commonality of our beliefs in future conformity to a convention [15], the salience of silence might justify a mutual belief in the occurrence of confirmation.

One relevant consequence of such common knowledge is that, though at the beginning Bob were just 'passively' allowing Alice to believe something about him, under these conditions of common knowledge of his confirmation, the allowing becomes 'active'.

Moreover, given what we have discussed in Section 5, this is sufficient for the Principle of Reliability to apply, giving rise to Bob's duty of reliability and to Alice's corresponding right to rely. And from this it follows that her possible protest or resentment cannot but be *entitled* simply because she has a *right* that he does as expected, that is, that he is reliable.

8 Tacit agreements: when the agreement is implicated

Even if agreements can be established without promises, usually other kinds of speech acts are employed to create the required epistemic conditions behind them. For instance, for an agent to consent another one to something that is desired, the former needs to know about such desire in the first place. Usually, the latter communicates the desire simply by informing, or by formulating a request or, sometimes, by proposing an exchange, and so aiming to offer a reason to motivate the former acceptance. A conditional promise is first of all a way to influence such acceptance by offering some incentives. Similarly, the consent must be mutually known between the parties, and, to this end, one's the intention not to interfere with the other's desire fulfilment is usually communicated. This is often done through explicit communication, that is, by conventionally signalling one's agreement through nodding or using verbal communication.

However, I can inform you about my desire just by taking the keys of your car, knowing that you are looking at me, and that you will infer the desire behind my behaviour. Analogously, by acting in order to remove an obstacle for me or by avoiding creating one, you can communicate with me without language, gestures or other conventional means. In fact, practical actions (or forbearances) done with a communicative intention (i.e. practical actions done also because another agent while 'reading' such behaviour will believe something) might suffice to send a message. Elsewhere, we have argued for the importance of this kind of communication that we name 'behavioural implicit communication' [2] [30]. Here, we just confine ourselves to suggest that this form of communication through practical actions and their effects might support the creation of agreements that can be dubbed, for this reason, *implicit agreements.* When there is an implicit agreement between some agents, the one having the power to interfere with the other can implicitly give his consent by acting with the intention to refrain from interfering, knowing that the other understands what is happening. Those that are qualified as 'tacit' are often instances of agreements established, silently, via implicit communication.

Notwithstanding so, if there are cases in which it is already common knowledge between the agents that one of them wants something, even implicit communication may be useless; similarly for the consent, the uptake, and all the conditions that need to be commonly known for an agreement to be in place.

But how is it possible that all these epistemic conditions be satisfied, without either promises or any other kind of communication between the parties? Or, in other words, *how is it possible to have agreements without communication?*

Recall the necessary and sufficient conditions to have an agreement discussed in Section 4. One prominent clause is the so-called 'consent condition'. In the way it has been formulated, such condition does not require any communication. In fact, having another agent's consent just entails that the agent with the power to interfere, indeed, intends not to interfere. However, often, one does not only consent to something, but one also *gives one's consent*, which necessarily is the communication of such decision of non-interference, via the usual Gricean mechanism [11]. One can give one's consent without verbal or gestural communication, but at least implicit communication is necessary. However, though one cannot be *given* the consent without communication, one can *have* the *tacit consent* without any communication.

Consider again the example discussed in the previous section.

It has been shown that when the parties consider silence as a confirmatory device for the beliefs on the truth of which one relies, the confirming agent becomes obliged to be reliable, even if no communication has occurred between them. In the example, Bob become obliged not to take the car tomorrow, given his tacit confirmation of a belief Alice had about him. However the mere fact of not taking the car, and as a consequence of not interfering with her is not in itself sufficient for Alice to have his consent. According to the analysis developed in Section 4, if one has a consent then the other agent has the intention not to interfere with him, that is, the consent implies that *the content of the intention refers to another agent*. Differently, the intention behind the behaviour that contingently happens not to create obstacles for another agent needs not be so. Indeed, in the example, the decision not to take the car on Monday is motivated either because that is Bob's habit on Mondays or because it is the best option he has to avoid being stuck in traffic.

However, as noted in Section 5, once the principle of Reliability applies, one incurs in a 'directed' obligation, rather than an unqualified one: Bob is *obliged towards* Alice not to take the car, and Alice has a *right against* Bob to this behaviour. Therefore, such a directed obligation is not simply to avoid taking the car, but, more precisely, to forbear to do what would, in this context, prevent her to fulfil her desire that Bob does not the car, which amounts to being obliged to not interfere with such desire fulfilment.

Granted this, is it true that Bob's silence also means that he intends not to interfere with Alice, i.e. that she has his consent?

Recall that Bob's silence is confirmatory of her belief about him under the presupposition of non-hostility; otherwise the disconfirmatory reading of the evidence would be maximally relevant. The presupposition that Bob desires not to be moved by a hostile attitude, however, amounts to assuming that the principle of Reliability is actually followed.

To see why it is so, consider Lewis' analysis of the kinematics of presuppositions in a conversation [19]. According to Lewis: "presuppositions evolves according to a rule of accommodation specifying that any presuppositions that are required by what is said straightway come into existence, *provided that nobody objects*" [19]. Though presuppositions are almost always approached in the contest of communication, the fact that social interaction, even tacit as in this case, may have the same properties and consequences of linguistic exchanges and proper conversations is explicitly endorsed by some pragmatists [14]. If a presupposition of reciprocal non-hostility, possibly grounded in a shared value of not being motivated by hostile attitudes, is reasonable, then what is required 'by what is done' when one is in a social interaction with another agent becomes immediately into existence. That is, it becomes common knowledge between the agents that both of them share a value *de se* not to be motivated by hostile attitudes. In the present context, Bob's violation of the principle of Reliability would amount to actively allowing that Alice incurred into losses, and he would be indeed hostile towards her. If we accept that there is such a presupposition of non-hostility in the background of this kind of interactions, then we are also accepting that there is a shared assumption between the agents that principle of Reliability is indeed followed. Under these conditions, and given that Bob's silence confirmed the belief that he will not take the car, and that he is consequently obliged not to interfere with her, his silence *also* means that he intends not to interfere with Alice or that she has his consent that her desire is fulfilled. More precisely, *if in this context one's silence "naturally" means one's confirmation* [11], *it also "implicates" one's consent* [12]: that an agent intends not to interfere with another one or that the latter has the consent of the former is an "implicature" of such tacit confirmation because it is required that the former agent has such an intention in order to preserve the shared assumption that he is not hostile towards the other, or, which is the same, that he is not violating the principle of Reliability since this is something that the latter is assuming the other is not doing[15].

To sum up, given a shared assumption of non-hostility and thanks to the process of tacit confirmation, Alice knows that Bob also has a sufficient reason, a normative reason, for consenting her to something that she wants, that is, he desire that Bob does not take the car this Monday. Under the same assumption of non-hostility, which in this context amounts to the assumption that the principle of Reliability is followed, she also has reason to believe that Bob intends not to interfere with her since, by being silent, he implicates that she has his consent. Moreover, given that the assumption of non-hostility is shared by the agents, and that both the tacit confirmation and the normative consequences are common knowledge, it is also commonly known that Bob's silence means (implicates) his consent. It is this kind of consent that we consider a *tacit consent*, that is, consent without any communication between the parties, which is tacit in the sense that is *implicated* by something your are doing and from what is already commonly known and assumed by the agents. As a consequence that Alice has such tacit consent is also commonly known without having been manifested in any way, that is, without Bob giving it to her.

Let's now consider conditions 2 and 3: the valuing one's power and the no coercion conditions.

An agreement between them that Bob does not take the car entails also (1), that he desires that it is his desire to use or not to use the power over her to move him to act and (2) that Alice acknowledges this power over her as far as this issue is concerned, that is, she intends not to oppose Bob's decision to interfere with her desire fulfilment.

However, there has been no deliberation to consent her to something in the first place, and Bob's tacit consent is just implicated by something he did. So, how can such consent be compatible with Bob valuing his power?

This is the same objection put forward by Hume against Locke's famous justification of political authority. Hume in fact in his *Of the Original Contract* has resisted the claim that such authority is the product of a tacit consent whereby "the subjects have tacitly reserved the power of resisting their sovereign" on the account that, "an

[15] 'Implicatures', like presuppositions, are usually approached in the context of conversation, a situation in which we use language for common aims in a way that, as Grice has suggested, is governed by a Cooperative Principle. However Grice notoriously claimed also that the principle and the related maxims apply to cooperative contexts that are not communicative [12]. The relation between Grice's Cooperative Principle and the weaker principle of Reliability exceeds the scope of this contribution and are left for future research.

implied consent can only have place, where a man imagines, that the matter depends on his choice", that is, where a man imagines that by desiring to interfere, he would thereby have successfully exercised his power. Whether this is so in relation to political authority is not of our concern here, but still for an agreement to be in place such condition, or better, conditions 2 and 3 of an agreement must be met.

We have argued above that the consent is normatively required by the fact that Bob has actively allowed Alice's reliance. Even if it is required, this does not mean that the consent has been coerced or that no other alternative was indeed possible. In fact, *if he had not confirmed her belief, she would have accepted his decision to act in ways that interfered with her desire fulfilment.* The truth of this counterfactual, together with the fact that Bob has indeed confirmed her belief about him are also sufficient to guarantee that, though she does acknowledge his power over her in this context, she is now entitled to fulfil her desire. But how can the agents mutually know that such a counterfactual is true of them?

Simply because the shared assumption of non-hostility requires it too. Suppose in fact that Bob thought differently. Bob imagines that even in case he hastened to disconfirm Alice's belief, she would have pursued her desire in any case. This belief is incompatible with the truth of proposition that Alice values non-hostility as much as Bob does. Given that there was indeed an alternative to what has happened (Bob could have disconfirmed her belief but he didn't) Bob has to assume, if the shared assumption is to be considered true, that she would have behaved in non-hostile way. Hence, both conditions 2 and 3 are also satisfied, or better implicated, by what it is already common knowledge between them.

Moreover since both the fact that he is moved by a desire not to interfere with her and that she acknowledges his power are implicated on the background of what they already commonly know, both conditions are common knowledge, or at least potentially so.

Finally, for the social relation between the agents to qualify as an agreement, as already argued, the agent having the consent needs to uptake it (condition 4) and this fact must be common knowledge between the parties.

At first glance it may seem that this condition is already established because, in the example, Alice is in fact already relying on Bob not taking the car tomorrow. However, the uptake of an agreement is not just reliance on another's behaviour that happens not to interfere with one's desire but is, more specifically, reliance on the other's *intention* not to interfere with such desire fulfilment (see Section 4); to have an agreement one does not merely rely on another's behaviour, one relies on an intention, that is, one uptakes a consent.

Since however, in the example, condition 1 is satisfied, Alice also has the opportunity to rely on his intention not to interfere with her desire fulfilment, and not simply on his observable behaviour. But how can such uptake on her part be common knowledge between them?

Suppose that she does not in fact uptake the tacit consent. She can do this for, at least, two very distinct reasons[16]. She can consider that he is not trustworthy enough, in the sense that, though he now desires not to interfere with her, she believes that he will indeed change his mind on this issue. Differently, despite the fact that Alice be-

[16] We thank Maria Miceli for clarifying the relevance of this distinction.

lieves in Bob's trustworthiness, she simply does not want to take his car anymore: it is Alice who has changed her mind. Both state of affairs are however incompatible with the shared assumption of non-hostility. Let's consider the latter first. If eventually Alice does not desire to take his car anymore, then, since he has decided not to interfere with her, Bob will in incur into losses (i.e. the opportunity costs Bob has already paid) given that he is relying on the fact that she has this desire. In fact, just as Bob's silence, her silence too is a continuing confirmation of a belief of his: the expectation that she still desires something from him. Thus, she has also actively allowed him to rely on something and, as a consequence, he has now acquired a right to the truth of this proposition, for the same reasons discussed above. If it is now too late for a warning, either she ought to compensate for the losses or she ought to fulfil his expectation, that is, Alice has to keep her desire in agreement with Bob's. Thus, her silence, like his, has in this context a natural or salient meaning: it means a confirmation that Alice still desires what Bob expects her to desire. On the other hand, given that both agents are presupposed to value non-hostility, Alice possible distrust in Bob is incompatible with his actual being non-hostile because by believing that he *will* change his mind, she would also believe that he *will* be hostile with her. And this is something that is ruled out by our shared assumption, or at least, it is something that is to be considered as false in order not to violate it. As a consequence, if Alice's silence naturally means that she still desires what he expects her to desire, and having common knowledge of the tacit consent, then Alice's silence means also, or better implicates, that she relies on his consent. This is what is implicated in order not to violate the shared assumption of non-hostility. Because this fact follows from something we already commonly know and assumed, it is again something that we commonly know.

Let's take stock. Though agreements are very often based on communication, there is a kind of agreement that is not based on any form of communication, not even implicit. It is for this kind that we reserve the name of *tacit agreement*. Crucial for the establishment of tacit agreements is the fact that *there is a salient interpretation for one's silence when it is common knowledge that an agent reasonably expects and wants something from another one or has a right to obtain*. It is due to the salience of silence as a confirmatory device that we tacitly, and often involuntarily, become obliged to be reliable. To account for such normativity the *prima facie* plausibility of a principle of Reliability has been invoked. Under a presupposition that the agents share a value *de se* of not being moved by hostile attitudes, there is also an assumption that the principle of Reliability is actually followed. As a consequence a tacit confirmation also means one's tacit consent, or better, it 'implicates' such consent. Though implicated, such consent is not however coerced because it is also implicated that things could have been different, and this counterfactual possibility is matter of common knowledge. Finally, once an agent has another's consent, it is again the salience of silence that guarantees that the last condition for an agreement is satisfied, that is, those who have the tacit consent tacitly confirm that they keep their desires in agreement and, on this basis, implicate their uptake. Tacit agreements are agreements without communication, and are established necessarily by the tacit confirmation of the involved parties. Tacit agreements are potential agreements in the sense that there are reasons to believe that all the conditions for an agreement are fulfilled and this fact is accessible to the parties, at least if they bothered to think hard enough. Tacit agreements remain potential as long as everything goes smoothly, that is, for example, if the agent who is in fact tacitly consenting, also acts as expected for whatever reason. They become actualized and operative agreements when one, willing to act against what the tacit agreement mandates, cannot but acknowledge that the consent, the uptakes and all the other conditions do in fact hold, that is, cannot but assent that a real agreement is in place. Finally tacit agreements, as all agreements, create reciprocal obligations and rights in the parties entering into them to keep their desires in agreement, that is, after an agreement is in place no unilateral change of mind is legitimate anymore.

9 Conventions are tacit unconditional agreements

If, following Hume's suggestion, conventions are agreements, and given that conventions persist without the need of communication, they are agreements without communication, that is, tacit agreements.

Consider a convention to drive on the right sustained by an interest in avoiding collisions.

As we have proposed in Section 3, conventions are regularities of reciprocal trust, hence, in the example, agents in the population regularly rely on the others to drive on the right: everyone assumes that the other will drive on the right and acts accordingly, that is, he himself drives on the right. Given that a convention presupposes an agreement in desire for some ends (our agreement in desiring not to collide), the expectation of reciprocal reliance is a reason for everyone to rely on each other so that, in this way, also our desire for the means (each desire to drive on the right in order to avoid collisions) are in agreement too.

Trust, as we have suggested in Section 3, is a fundamental non-hostile attitude on the part of the trustor: an agent relies on another to do an action that stems from his motivation, without any coercion. The reason why each relies on the others when they are parties of a convention is that each one expects the others to rely on oneself in the same, non-hostile, way. Moreover in order to trust everyone has to assume such non-hostile attitudes in the trustees. Suppose, then, as we have done in Section 6 that the agents in the population share a value of not being motivated by hostile attitudes, something that, of course, would promote the disposition to trust each other. Suppose also, as we have done in Section 7, that the agents have a bias for confirmation.

Under these two assumptions, and given that a convention exists in a population, each time two or more agents interact with each other in a situation that is governed by the convention, if they keep silent about the expectation of reciprocal reliance that they mutually know to have, each of them confirms their reasonable expectations about each other, *even if their mutual expectations of reciprocal reliance are not grounded in direct experience*; the agents might have never met before. By being confirmatory, each actively allows reliance on the truth of such expectation of reciprocal reliance. As a consequence, each also acquires both a *right* that the other rely on oneself, and an *obligation towards the other* to rely on the other one. Each agent has now a right that the other drives on the right (i.e. has a right to be trusted) and an obligation to drive on the right himself (i.e. ought to trust the other one).

Moreover, for the same reasons discussed in Section 8, on the basis of a presupposition of reciprocal non-hostility, each silence also “implicates” each consent, that is, that each intends not to interfere with the desire fulfilment of the other one. Given that in a convention, all the agents desire conformity of all the others, the tacit consent is the decision not to interfere with this desire of one’s own conformity. And since conformity of others to a convention amounts to that fact that the others do rely on oneself, in the example, one’s silence implicates one’s tacit consent to all the others that one has decided not to interfere with their desires to rely on them. In a convention, each also tacitly consents to trust the others.

Moreover, in any convention it is the individual interest of each agent to conform, that is, everyone trusts the others because it is in the interest of everyone not to collide with the others, and so to rely on the others by driving on the right. Everyone’s desire for the means stems from everyone’s motivation not to collide. This very basic capacity (or power) of instrumental rationality is something that everyone values and everyone acknowledges to the others. If one had known that was not in the interest of the others to drive on the right, that is, to rely on oneself, one would have acted accordingly. This much is granted both by the fact that the agents are in a coordination problem [15], and in order to preserve our presupposition of non-hostility.

Finally, each uptakes such tacit consent of the other by tacitly confirming, firstly, that others’ trust on oneself is still something one desires, and, secondly, by implicating that one does rely on such trust on oneself of the others and will act accordingly. That the uptake holds is required again by the presupposition of non-hostility, and has the consequence that each does not only trust the others, but also rely on the trust of the others on oneself.

Each time the agents, ignorant of each other’s identities as they may be, do meet and keep silent about each other mutual expectations of reciprocal reliance establish or implicate a *tacit agreement to trust each other*. Since the tacit agreement is implicated by one’s own silence *both as a trustor and as a trustee* the agreement is reciprocal: there is a tacit agreement between the interacting agents the both trust and are trusted by the other one. The tacit agreement is unconditional because the tacit consent are not conditioned one on the other; differently they are implicated by the presupposition that the agents are non-hostile, or, in the specific context, that the principle of Reliability is followed. Finally, the normativity of conventions is that of the tacit or implicated agreement: by tacit agreeing to trust each other *everyone is obliged to keep one’s desires for the means in agreement with the other and has a right that the others do the same*.

10 Why conventions are tacit agreements

A regularity is a convention for the way it persists, not for its origins. In convention, one conforms if the others conform because it is in one’s interest to conform. Since the stability of conventions is guaranteed by this specific motivational structure (i.e. their pre-existing agreement in desiring some end) together with common knowledge of all the conditions specified in Section 2, individual instrumental rationality alone suffices to stabilize it. Then, why should a convention be also a tacit agreement? Isn’t is only just an additional pressure that is made redundant by the reasons the agents already have for acting as they do? What is the role of obligations and rights in conventions?

Though it is true that conventions are stable for these reasons, the fundamental condition that ensures stability is that the agents agree in desiring jointly co-realizable ends. But what is there to guarantee that they will keep doing so? After all a common interest needs not be some ultimate end that we will invariably pursue forever. The ends we agree in desiring are often just means for some further ends we have. All instrumental desires cease to be motivationally effective, once the end in light of which we pursue the means has been either fulfilled or abandoned. Suppose Alice and Bob have a common desire to meet each other one day during the week and they fulfil their desires following the convention to go at the movie together every Wednesday. Suppose also that Bob is secretly in love with Alice, and hopes that by recurrently meeting him she will fall in love too. Differently, for Alice, Bob is just a friend that she is keen to meet, and nothing more. This Wednesday, at the end, Bob realizes how desperate his situation is, how impossible it is that his love will be ever reciprocated, and he abandons his plan to seduce Alice altogether. If he suddenly revised his recurrent end to meet with Alice, there would no motive at all to still pursue the means of going to the movie with Alice that night. Still however, by not showing up, Bob would do something wrong and against Alice, something that, notwithstanding his feelings, he may wish to desire not to be moved to do.

In other words, since all the parties to a convention conform (trust) on the assumption of the trust of others, agents need protection and assurance against the mutability of interest that might compromise each individual project. Since the kind of common interest presupposed by a convention may be as volatile as any other end we pursue, everyone would be at risk if everyone were free to change one's mind without taking into account the other in any way. Obligations act as further assurance in case one was to change his desires by entitling possible influencing actions (e.g. punishment by reproach), which can motivate the others beside their current desires.

Conventions tend to reproduce agreement in desiring arbitrary means from agreement in desires for the ends. However, by also being sources of tacit agreements between the agents, the arbitrary means are turned into ends to be pursued unless one is able to warn the other in time or is prepared to compensate for possible losses.

11 Conclusion

In his paper on causation, Lewis noted that Hume has defined a causal succession "twice over" [16][17]. The aim of this article is to suggest that something similar has occurred when Hume defined a convention as: "a general sense of common interest, which sense all the members of society express to one another, and which induces them to regulate their conduct by certain rules. […] When this common sense of interest is mutually expressed, and is known to both, it produces a suitable resolution

[17] Hume defines a causal succession both as a succession that institutes a regularity and by way of a counterfactual analysis. The two notions are to be kept separated, see Lewis (1973).

and behaviour. And this may properly enough be called a *convention or agreement betwixt us*, *though without the interposition of a promise*; since the actions of each of us have a reference to those of the other, and are performed upon the supposition, that something is to be performed on the other part" (Hume, *A Treatise of Human Nature*, III.ii.2, emphasis added).

That convention can be seen as tacit agreements is often suggested, and is considered as tantamount to the analysis offered by Lewis. However, what Lewis has shown is that, in certain conditions, an agreement in desires for the means might stem from our independent agreement in desires for the ends. However an agreement in desires is not the same as an agreement between the agents in that the latter, but not the former, is a social relationship between the agents. The fact the there is an agreement between the agents entails that their relationship is also a normative relationship. Whereas their mere agreement in desires may not have such consequences.

In this paper we have shown that the normativity of conventions is the normativity of tacit agreements, that is, that the agent becomes bound to keep their desires for the means in agreement, and by becoming so bound they are assures the other will not change their minds without some concern for their fellows.

The agreements that stem from conventions are tacit in the sense that they are implicated by what the agents do (or forbear to do) though without any communication between them is necessary. In order for this be possible we have offered two substantial hypotheses: (1) that there is a salient interpretation, in some contexts, of everyone's silence as confirmatory of the others' expectations, and (2) that the agents share a value of not being motivated by hostile attitudes, ad, on this basis that their interaction are regulated by a presupposition that the principle of Reliability is followed. If the former hypothesis is compatible with many available empirical data about human decision-making (Section 7), the latter is matter of future research.

References

1. Bacharach, M. & Gambetta, D. (2000) Trust in signs. In Karen Cook (Ed) *Trust and Social Structure*. New York: Russell Sage Foundation.
2. Castelfranchi. C. (2006) From conversation to interaction via behavioral communication. In S. Bagnara and G. Crampton-Smith (Eds.*) Theories and Practice in Interaction Design*, pp. 157-179. New Jersey (USA): Erlbaum.
3. Castelfranchi, C. & Falcone, R. (in press) *Trust Theory: Structures, Processes, Dynamics*. Wiley & Sons.
4. Conte, R. & Castelfranchi, C. (1995) *Cognitive and Social Action*. London: UCL Press.
5. Ellsberg, D. (1961) Risk, ambiguity, and the savage axioms, *The Quarterly Journal of Economics*, 75(4), pp. 643-669.
6. Evans, J. St. B. T. (1989). *Bias in human reasoning: Causes and consequences*.. Hillsdale, NJ: Erlbaum.
7. Frankfurt, H. (1971) Freedom of the Will and the Concept of a Person, *Journal of Philosophy*, 68, pp. 5-20
8. Gilbert, M. (1983) *On Social Facts*. London and New York: Routledge.

9. Gilbert, M. (1993) Is an agreement an exchange of promises?, *The Journal of Philosophy*, 54 (12), pp. 627-649.
10. Gilbert, M. (2008) Social convention revisited. *Topoi*, 27(1-2).
11. Grice, P. (1957) Meaning, *Philosophical Review*, 66, pp. 377-388.
12. Grice, P. (1989) *Studies in the Ways of Words*. Cambridge (MA): Harvard University Press.
13. Holton, R. (1994) Deciding to trust, coming to believe, *Australasian Journal of Philosophy*, pp. 63-76.
14. Levinson, S.C. (1995) Interactional biases in human thinking. In E. Goody (Ed.) *Social Intelligence and Interaction*, pp. 221-260. Cambridge: Cambridge University Press.
15. Lewis, D. (1969) *Convention: A Philosophical Study*. Cambridge (MA): Harvard University Press.
16. Lewis, D. (1973) Causation, *Journal of Philosophy*, 70, pp 556-67.
17. Lewis, D. (1975) Languages and Language. In K. Gunderstone (Ed.) *Minnesota Studies in the Philosophy of Science*, vol. VII, University of Minnesota Press (reprinted in his Philosophical Papers, volume 1, pp. 163-188).
18. Lewis, D. (1979a) Attitudes de dicto and de se, *The Philosophical Review*, 88(4), pp. 513-543.
19. Lewis, D. (1979b) Scorekeeping in a language game, *Journal of Philosophical Logic*, 8, pp. 339-359.
20. Lewis, D. (1989) Dispositional Theories of Value, *Proceedings of the Aristotelian Society, Supplementary Volumes*, 63, pp. 113-137.
21. Marmor, A. (1996) On convention, *Synthese*, 107, pp. 349-371.
22. Miceli, M. & Castelfranchi, C. (2002) The mind and the future. The (negative) power of expectations, *Theory & Psychology*, 12(3), pp. 335-366.
23. Nickerson, R.S. (1998) Confirmation bias: a ubiquitous phenomenon in many guises, *Review of General Psychology*, 2(2), 175-220.
24. Postema, G.T. (1982) Coordination and convention at the foundation of law, *The Journal of Legal Studies*, 11(1), pp. 165-203.
25. Rabin, M. & Schrag, J.L. (1999) First impressions matter: a model of confirmatory bias, *The Quarterly Journal of Economics*, 114(1), pp. 37-82.
26. Robins, M. (1984) *Promising, Intending and Moral Autonomy*. Cambridge: Cambridge University Press.
27. Savage, L. J. (1954) *The Foundations of Statistics*. New York: John Wiley & Sons.
28. Scanlon, T. (1990) Promises and Practices, *Philosophy and Public Affairs*, 19, 199-226.
29. Thomson, J.J. (1990) *The Realm of Rights*, Cambridge (MA): Harvard University Press.
30. Tummolini, L. & Castelfranchi C. (2007) Trace signals: The meanings of stigmergy. In D. Weyns, V. Parunak, and F. Michel (Eds.) *Environments for Multi-Agent Systems III*, number 4389 in Lecture Notes in Artificial Intelligence, pp. 141-156, Berlin/Heidelberg: Springer-Verlag.
31. Watson, G. (1975) Free agency, *Journal of Philosophy*, 72, pp. 205-220.

A Conviviality Measure for Early Requirement Phase of Multiagent System Design

Patrice Caire[1], Leendert van der Torre[1]

Computer Science and Communication, University of Luxembourg, Luxembourg

Abstract. In this paper, we consider the design of convivial multi-agent systems. Conviviality has recently been proposed as a social concept to develop multi-agent systems. In this paper we introduce temporal dependence networks to model the evolution of dependence networks and conviviality over time, we introduce epistemic dependence networks to combine the viewpoints of stakeholders, and we introduce normative dependence networks to model the transformation of social dependencies by hiding power relations and social structures to facilitate social interactions. We show how to use these visual languages in design, and we illustrate the design method using an example on virtual children adoptions.

1 Introduction

The focus of this paper is the social/organizational structure of a multiagent system. In particular, we are interested in the design of *convivial* multiagent systems, which is directly related to well studied issues such as groups and teams, norms and normative behavior, and coalition formation. First, we discuss the determining factors and the decisions we have to make concerning the actual convivial characteristics of the system. Following the TROPOS methodology, this process leads us to our dependence network model. A crucial step in this phase is to manage conflicting requirements such as reconciling freedom with exclusion and missing or incomplete specifications such as implicit agents goals. Second, we propose a representation of our model and present our formalism, initially expressing dependencies with static dependence network. We then express the sequence of different actors point of views, temporal dynamic networks. Third, we define the actors interactions and model a protocol.

We study the following research questions:

1. How to design the evolution of convivial social relations?
2. How to combine viewpoints from stakeholders?
3. How to incorporate normative aspects of conviviality?

The description level of this paper is methodologies and languages. To answer these questions we develop temporal dependence networks to model the evolution of dependence networks and conviviality over time, we introduce epistemic dependence networks to combine the viewpoints of stakeholders, and we introduce normative dependence networks to model the transformation of social dependencies by hiding power relations and social structures to facilitate social interactions.

Dagstuhl Seminar Proceedings 09121
Normative Multi-Agent Systems
http://drops.dagstuhl.de/opus/volltexte/2009/1899

The inspiration source of our work is political and social science. Empathy and reciprocity were foregrounded by Polanyi in 1964. “Individual freedom realized in personal interdependence" was tooled up by Illich in 1974 [17]. And in 1988, Putnam considered conviviality as a condition for civil society and social capital, a concept referring to the collective values of all social networks. One of the four themes of the European Community fifth framework program was entitled the “societe de l’information conviviale” (1998-2002) [25], which was translated as “the user-friendly information society.” Today, a number of research fields such as computer supported cooperative work and social software aim at supporting users to interact and share data. Conviviality has recently been proposed also as a social concept to develop multi-agent systems [9].

As a running example, we use the design of a virtual adoption agency for instance on Second Life (SL). Adopting virtual children is a successful experience and a flourishing business on SL. Parents wishing to adopt a child must pay a fee to the adoption agency. The procedure typically involves that parents list themselves to advertise their profile to prospective children who can select them. The agency then matches children and parents and organizes a try-out period. There is no pressure. Once parents and children have made their decision, they simply come back to the agency to cancel the adoption if unhappy or otherwise to confirm it and get their adoption certificate and a ceremony. The experience must be convivial.

The conviviality literature discusses many definitions and relations with other social concepts, which we do not introduce in the formal model in this paper, referring to qualities such as trust, privacy and community identity. Also, in this paper we do not consider Polanyi’s notion of empathy, which needs trust, shared commitments and mutual efforts to build up and maintain conviviality.

The layout of this paper is as follows. In Section 2 we discuss the social focus of this paper by explaining how the social concept “conviviality” can be used to develop multiagent systems in general, and their design in particular. In the following four sections we answer the research questions. In Section 3 we introduce temporal dependence networks to model the evolution of dependence networks and conviviality over time. In Section 4 we introduce epistemic dependence networks to combine the viewpoints of stakeholders. In section 5 we introduce normative dependence networks to model the transformation of social dependencies.

2 Convivial multiagent systems

In this section we discuss the use of social concepts in general, and “conviviality” in particular, for the development of multiagent systems.

2.1 Social concepts in multiagent systems

A social concept like “conviviality” can be used in multiagent systems in various ways. Consider the following examples:

Informal requirements of decision makers: “our system should be convivial and easy to use”

Formal concept in an ontology for modeling multiagent systems: “system A is convivial whereas system B is efficient”
Performance measures: “the conviviality is 87 on a scale from 0 to 100”
Programming constructs: “if use$<$10 then conviviality++”

Though the latter ones may seem farfetched at the moment, consider some of the many other social concepts have been adopted by computer science at all these different levels, from concepts in informal requirements via modeling concepts in UML to programming constructs (this list is far from complete!).

“Service” is a concept from business economics which has been used in computer science in service oriented architectures and in web services. Not only business processes but also computer applications are modeled as service providers.
“Contract” has been introduced in Meyer’s design by contract [19, 18, 1], a well known software design methodology that views software construction as based on contracts between clients (callers) and suppliers (routines), relying on mutual obligations and benefits made explicit by assertions.
“Coordination” is emerging as an interdisciplinary concept to deal with the complexity of compositionality and interaction. Coordination languages, models and systems constitute a recent field of study in programming and software systems, with the goal of finding solutions to the problem of managing the interaction among concurrent programs.
“Trust” and reputation are used as fundamental concepts in security.
“Architecture” is defined as the fundamental organization of a system embodied in its components, their relationships to each other, and to the environment, and the principles guiding its design and evolution. The recent standard called IEEE 1471-2000 [2] emphasizes that views on the architecture should always be considered in the context of a viewpoint of a stakeholder (e.g., software engineer, business manager) with a particular concern (e.g., security).
Value and quality are economic concepts. Value networks model the creation, distribution, and consumption of economic value in a network of multiple enterprizes and end-consumers.

Concepts, models and theories from the social sciences are studied in multiagent systems to regulate or control interactions among agents [3], as a theoretical basis for the development of so-called social software [21], and to develop multi-agent systems for computational social science [10]. Examples of social concepts studied in multi-agent systems are societies, coalitions, organizations, institutions, norms, power, and trust [11].

2.2 Conviviality requirements

Requirements for multiagent systems say that systems must be convivial, whereas system researchers and developers use other concepts. To model the requirement, the developers may interpret the conviviality requirement as being autonomous to make suggestions, to react the discussion in the meeting to reach their goals, being pro-active to take the initiative and being goal-directed, and most importantly being social by interacting with others to reach their goals.

When writing down requirements for user friendly multiagent systems, it is crucial to understand the inherent threads of conviviality, such as deception, group fragmentation and reductionism [9]. Whereas conviviality was put forward by Illich as a positive concept, also negative aspects were discussed. People are often not rational and cooperative to achieve conviviality [23] and unity through diversity [16] may lead to suppression of minorities. Taylor explores the contradiction that conviviality cannot exist outside institutions: i.e., the question “whether it is possible for convivial institutions to exist other than by simply creating another set of power relationships and social orders that, during the moment of involvement, appear to allow free rein to individual expression. Community members may experience a sense of conviviality which is deceptive and which disappears as soon as the members return to the alienation of their fragmented lives.”

2.3 Conviviality ontology

The use of conviviality as a computer science concept ensures that considerations on the user-friendliness of multiagent systems get the same importance and considerations on the functionality of the system. For example, our experience with the development of a digital city in Europe is that computer engineers are focussed on filling in forms and developing menu structures and other interface issues, and do not take into account that a digital city should be a meeting place for human and artificial agents.

Conviviality is a useful high level modeling concept for organizations and communities, emphasizing the social side of them rather than the legal side. Erickson and Kellogg [14] say: “In socially translucent systems, we believe it will be easier for users to carry on coherent discussions; to observe and imitate others’ actions; to engage in peer pressure; to create, notice, and conform to social conventions. We see social translucence as a fundamental requirement for supporting all types of communication and collaboration". Taylor studies conviviality in British pantomime and observes that: “conviviality masks the power relationships and social structures that govern societies.”

2.4 Design of convivial systems

In this paper we study how convivial multiagent systems can be designed using our operationalized concept of conviviality. We illustrate our arguments and contributions with a running example on multiagent systems for virtual adoptions, where typically physical reality such as multiagent technologies interact with virtual and social realities.

The aim of social scientists to create conviviality by creating the desired conditions for social interaction, coincides with the aim of designers of multiagent systems. For example, Illich defines a convivial learning experience in which the teacher and the student switch roles, such that the teacher becomes the student and the student becomes the teacher. This role swapping emphasizes the role of reciprocity as a key component for conviviality. Parallelely the importance of reciprocity in conviviality was shown for instance in [15]. As a result, such role swapping scenarios can directly be used in multi-agent systems.

3 Temporal dependence networks

In this section, we propose a design methodology for convivial multi-agent systems based on the agent-oriented software development process, Tropos [4]. Key ideas in Tropos are first, that throughout the process phases, e.g. from early requirements to implementation, agents are endowed with intentionality. Second, the importance of very early phases of requirement analysis to allow for a profound understanding of the environment and of the interactions for the software to be built. This methodology guides designer through an incremental process, from the initial model of stakeholders, to refined intermediate models that, at the end, becomes the code.

3.1 Dependence networks

Multiagent systems technology can be used to create tools for conviviality. Illich defines conviviality as "individual freedom realized in personal interdependence" [17]. Dependence network is a tool that allows us to model this interdependence [11, 24]. In a recently published paper [9] dependence networks were formally defined as in Def. 1.

Definition 1 (Dependence networks). *A dependence network is a tuple $\langle A, G, dep, \geq \rangle$ where:*

- *A is a set of agents*
- *G is a set of goals*
- *$dep : A \times 2^A \rightarrow 2^{2^G}$ is a function that relates with each pair of an agent and a set of agents, all the sets of goals on which the first depends on the second.*
- *$\geq$: $A \rightarrow 2^G \times 2^G$ is for each agent a total pre-order on goals which occur in its dependencies: $G_1 \geq (a)G_2$ implies that $\exists B, C \subseteq A$ such that $a \in B$ and $G_1, G_2 \in depend(B, C)$.*

Nevertheless, this representation of conviviality is static and therefore has a limited field of application. In the next sub-section, we present our extension to encompass the temporal aspect of conviviality.

3.2 Temporal dependence networks

Before proposing our definition, we introduce our virtual adoption running example. The procedure typically involves that parents list themselves to advertise their profile to prospective children who, if they like the parents, can select them. The agency then matches children and parents and organizes a try-out period. Once parents and children have made their decision, they simply come back to the agency to cancel the adoption if unhappy or otherwise to confirm it and get their adoption certificate and a ceremony.

We start by informally listing critical stakeholders. We then identify the relevant goals and the social dependencies of the stakeholders represented as actors. In particular, the actor **Parent** is associated with the goal: adopt child, while the actor **Child** is associated with the goal: get adopted and **Virtual Agency** with the goal: provide adoption service.

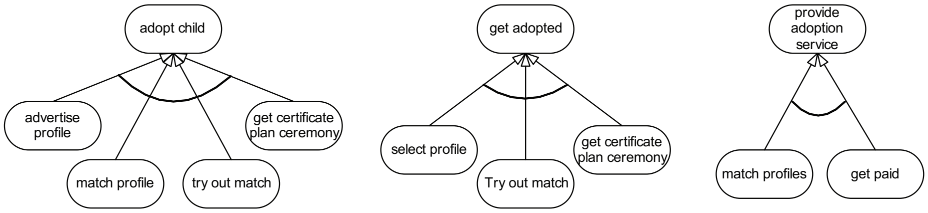


Fig. 1. Decomposition of goals.

To enrich the model with a finer goal structure and elicit dependencies, we decompose each root goal into sub-goals. For instance, **Child** goal: get adopted, is decomposed into three sub-goals: select profile, try out match and get certificate - plan ceremony. In Fig. 1, a graphical representation of goal modeling is given through a goal diagram; AND decomposition only are shown, no OR decomposition, e.g. no alternate sub-goals.

The UML sequence diagram (Fig. 2), illustrates the interactions among the stakeholders and how operations are carried out. The diagram shows time incrementing vertically. In particular, the diagram models the interaction among the three Users: **parent**, **agency** and **child**. The interaction starts with the advertise profile request by the **parent** to the **agency** and ends with the pay fee by the **parent** to the **agency**. We note that the match ok sent by both **parent** and **child** can be asynchronous. Moreover, the **agency** sends the adoption certificate and the plan ceremony to both **child** and **parent**.

Based on actor diagrams and goal decomposition, we proceed with a goal analysis taking each actor point of view. The objective is to obtain a set of strategic dependencies among the actors. We therefore perform an iterative analysis on each goal until all are analyzed. We build a succession of dependence networks from each actor point of view.

With temporal dependence networks, we aim at analyzing the evolution of dependence networks and conviviality over time. We identify the most relevant interactions in our running example and build a model with the key succession of dependence networks.

Definition 2 (Temporal dependence networks). *A dependence network is a tuple* $DP = \langle A, G, goals, dep \rangle$ *where:*

- A *is a set of agents*
- G *is a set of goals*
- T *is the set of natural numbers*
- $goals : T \times A \rightarrow 2^G$ *is a function that relates with each pair of a sequence number and an agent, the set of goals the agent is interested in.*
- $dep : T \times A \times 2^A \rightarrow 2^{2^G}$ *is a function that relates with each triple of a sequence number, an agent and a set of agents, all the sets of goals on which the first depends on the second if the third creates the dependency.*

We use this structure to model our example (Fig. 3). Note that the set of agents does not change, but the goals of the agents and the dependencies among them, changes over time.

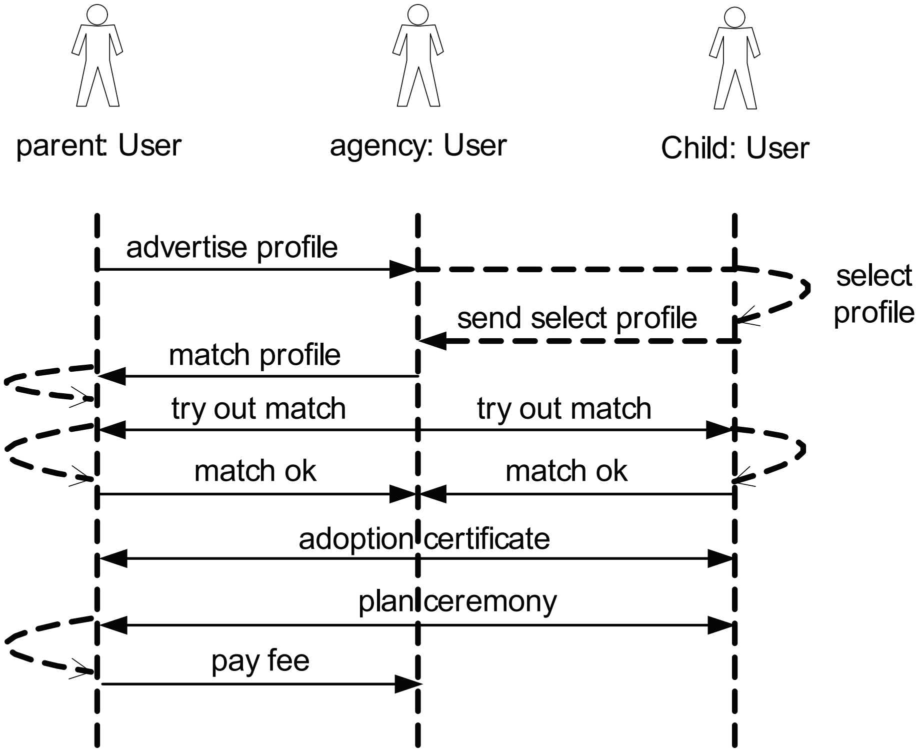


Fig. 2. Actor diagram modeling the stakeholders for the virtual adoption domain.

Agents $A = \{P, C, VA\}$ and
Goals $G = \{g_1, g_2, g_3, g_4, g_5, g_6, g_7, g_8, g_9, g_{10}\}$
We thus have the following sequence of dependence networks:
$DP_4 = \langle A, G, goals_4, dep_4 \rangle$, where:

- $goals(4, VA) = \{\{g_5, g_6, g_7\}\}$: In dep_4, the goals of agent *VA* are to provide adoption service, to get paid and to match parent-child profiles.
- $goals(4, P) = \{\{g_1, g_{10}\}\}$: In dep_4, the goals of agent P are to adopt a child and to try out match.
- $goals(4, C) = \{\{g_8, g_{10}\}\}$: In dep_4, the goals of agent C are to get adopted and to try out match.
- $dep(4, VA, \{P, C\}) = \{\{g_7\}\}$: In dep_4, agent *VA* depends on agents P and C to achieve goal g_7: match parent-child profiles.
- $dep(4, P, \{C\}) = \{\{g_{10}\}\}$: In dep_4, agent P depends on agents C to achieve goal g_{10}: try out match.
- $dep(4, C, \{P\}) = \{\{g_{10}\}\}$: In dep_4, agent C depends on agents P to achieve goal g_{10}: try out match.

In our notation, dep_i refers to the temporal dependence network where $i \in T$ and denotes the i^{th} sequence, P refers to agent **Parent**, C to agent **Child** and *VA* to agent **Virtual Agency**.

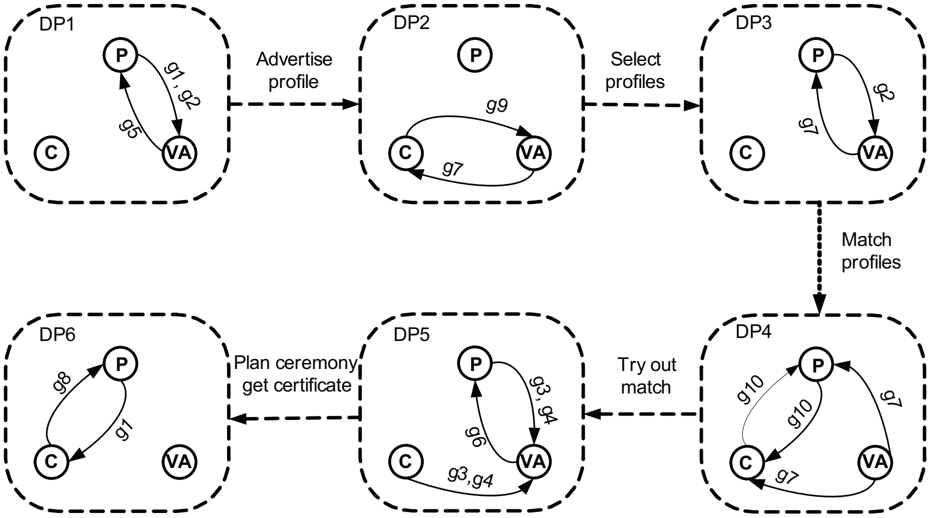


Fig. 3. DP sequences

4 Epistemic dependence networks

In our running example, we use the Tropos methodology [4], with the difference that we include neither plans nor resources. However similarly to Tropos, we identify actors which depend on each other to achieve their hardgoals, simply referred to as goals, and softgoals, the latter being typically used to model non-functional requirements and “having no clear -cut definition and/or criteria for deciding wheter they are satisfied or not" [4]. In Fig. 4, we show an *actor diagram* for the virtual adoption. In particular, **Parent** is associated with the goal: adopt child, and the softgoal: get nice child. Similarly, **Child** is associated with the goal: get adopted and the softgoal get nice parents while **Virtual agency** wants to provide adoption service and has the softgoal to provide a good service. Finally, the diagram includes one softgoal dependency where **Parent** depends on **Virtual agency** to fulfill the softgoal: adoption fee well spent.

Temporal dependence networks allow us to capture a relation from a specific point of view and at a specific time. Unfortunately, it is not sufficient for the situation we want to model, so in the next section, we try to answer this question by introducing a new model that will allow us to capture a more global view from the system point of view.

In order to model such system, we use the epistemic dependence network formally defined as Def. 3.

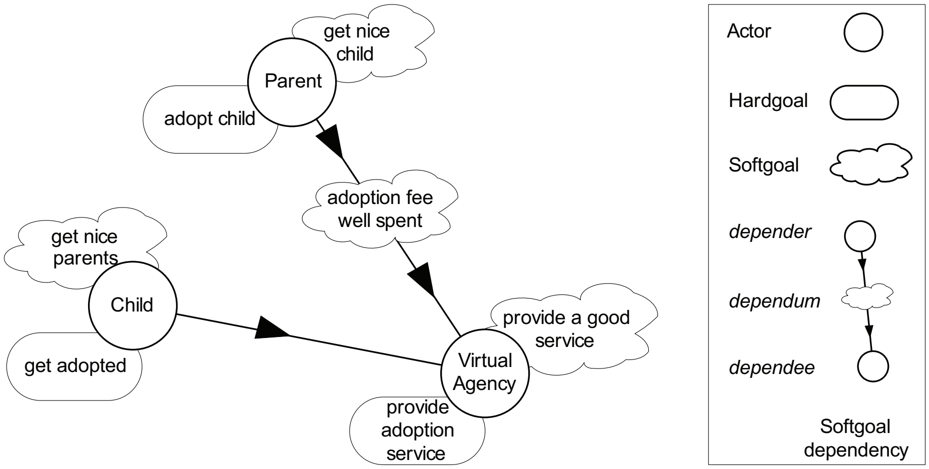


Fig. 4. Actor diagram modeling the stakeholders for the virtual adoption.

Definition 3 (Epistemic dependence networks). *An epistemic dependence network is a tuple $DP = \langle A, G, T, goals, dep \rangle$ where:*

- *A is a set of agents*
- *G is a set of goals*
- *T is the set of natural numbers*
- *$goals : T \times A \rightarrow 2^G$ is a function that relates with each pair of sequence number and an agent, the set of goals the agent is interested in.*
- *$dep : A \rightarrow T \times A \times 2^A \rightarrow 2^{2^G}$ is a function that expresses from the point of view of an agent $a \in A$, the dependence relation between another agent $b \in A$ and a set of other agents regarding the goals of agent b in a sequence $t \in T$.*

If we consider Fig. 5 the starting *goal diagram*, the three steps of this design process are:

1. Goal delegation: Each goal of any actor may be delegated to any other actor, already existing or new. It proceeds with the analysis of goals from the point of view of each actor. This generates a network of delegation between stakeholders, external actors and the system. The inclusion of new actors and sub-actors and subsequently, the delegation of sub-goals to sub-actors continues until all goals have been analyzed. Actors that contribute to the requirements are also included.
2. Goal decomposition: Goals and softgoals are further decomposed into sub-goals or found not reachable. Through this refinement process a goal hierarchy is created where leaf goals represent alternatives to root goals. Moreover, some identified sub-goals become reasons for new dependencies with new actors. Therefore, dependencies in actors diagrams must be revised.
3. When all actors fulfill their goals, all the goals have been analyzed and the root goals are satisfied then, this design process is complete.

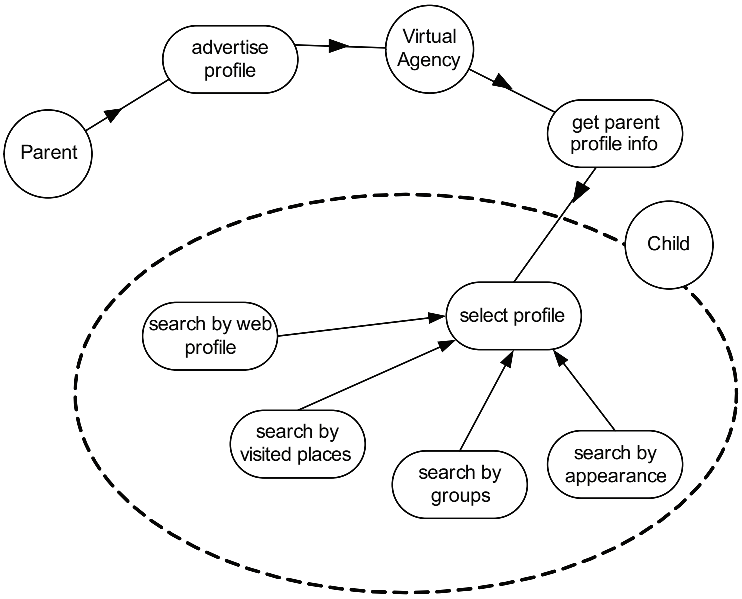


Fig. 5. Goal diagram for the goal select profile and dependencies between the actor Child and other environment's actors.

4.1 Example

In our running example, let's consider the set of agents

$A = \{P, C, VA, AS\}$, where *AS* is the **Adoption System**.

$dep(P) = (2, VA, \{C\}) = \{g_9\}$: **Parent** believes that in sequence 2, **Adoption System** depends on **Child** to achieve goal g_9: select profile.

We express Fig. 6 as follows: $dep(AS) = (2, P, \{C\}) = \{g_9\}$: **Adoption System** believes that in sequence 2, **Parent** depends on **Child** to achieve goal $\{g_9\}$: select profile. We note that there is no dependency from **Adoption System** towards **Adoption System** for the goal: select profile.

With Fig. 5 and 6, we explain the iterative design process from the Tropos methodology that are tool supported [22].

To explain what is the delegation process, and as an example, we here give a partial view on goal: select profile.

To start, we have the goal of **Child**: select profile. After analyzing the rational for this goal from each actor point of view, we delegate this goal to the new actor, the system-to-be **Adoption System** . We continue by analyzing each sub-goal.

We then identify the capabilities needed by **Adoption System** to fulfill all the four identified sub-goals: search by web profile, search by visited places, search by groups and search by appearance. In order for this latter sub-goal to be fulfilled, we add a new goal: provide photo/video and a new dependency from **Adoption System** towards **Parent**. Similarly, in Fig. 5 the sub-goal: search by web profile has no dependency while in 6 a new dependency from **Adoption System** towards **Child** has been created to fulfill the subgoal: know web address. Of course, each dependency must be mapped to a capability. We then define a set of agent types and assign each of them one or more capabilities. The specification of agent's goals, beliefs, capabilities and the communication between the agents depends on the adopted platform and the chosen programming language. We therefore leave this part for further work.

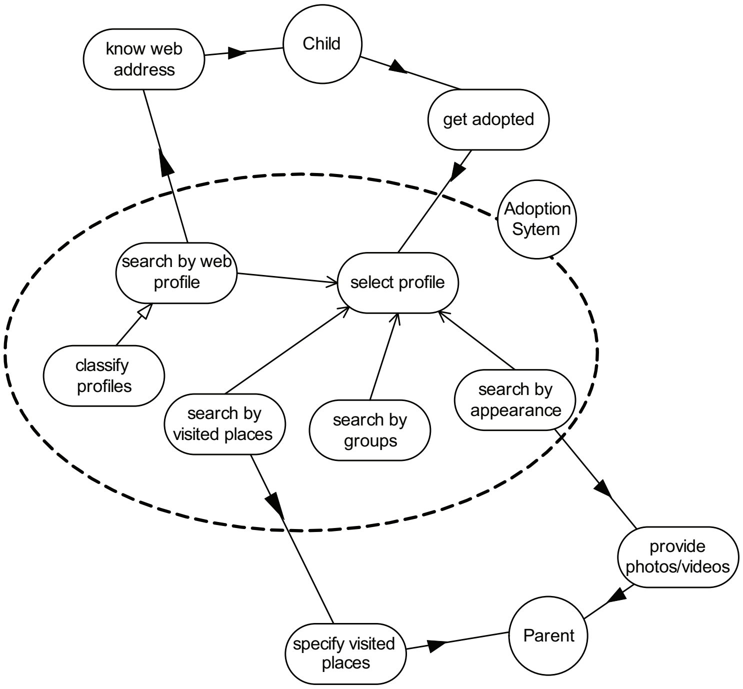


Fig. 6. Goal diagram for the goal select profile and dependencies between the actor Adoption System and other environment's actors.

4.2 Nested dependencies

We first mention that by *nested* we simply mean a belief produced and only accessible by an agent a and about another agent b, e.g. inaccessible to all others. For instance, empathy provides a way to know what another agent's preference is, and therefore to better adapt to it, allowing for a convivial relation, whereby agents contribute to each other. In our running example, let's assume that **Parent** believes that **Child** depends on it, **Parent**, for its goal: select profile. Let's further assume that **Child** believes that **Parent** depends on it to advertise parent profile, for example if **Child** first had to publish an announcement on a board that it is seeking parents to be adopted by. We write:

$dep(P) = (1, C, \{P\}) = \{g_9\}$: agent P believes that in sequence 1, agent C depends on it, P to achieve its goal g_9: select parents' profile.

$dep(C) = (1, P, \{C\}) = \{g_2\}$: agent C believes that in sequence 1, agent P depends on it, C, to achieve its goal g_2: advertise its profile.

5 Norms and masks

There are many different kinds of goals, some goals may be considered normative, others personal. Agents do not only have personal goals, they also have normative goals, e.g. goals imposed by the procedures. We propose a further extension of epistemic dependence networks that we call "Normative epistemic dependence networks" in order to take into account the differences in the two kinds of goals as well as obligations and violations.

Definition 4 (Normative epistemic dependence networks). *A dependence network is a tuple*
$DP = \langle A, G, N, O, V, T, goals, dep\rangle$ *where:*

- A *is a set of agents*
- G *is a set of goals*
- N *is a set of norms*
- T *is the set of natural numbers*
- $O : N \times A \rightarrow 2^G$ *is a function that associates with each norm and agent the goals the agent must achieve to fulfill the norm; We assume for all* $n \in N$ *and* $a \in A$ *that* $O(n, a) \in power(\{a\})$*;*
- $V : N \times A \rightarrow 2^G$ *is a function that associates with each norm and agent the goals that will not be achieved if the norm is violated by agent* a*; We assume for each* $B \subseteq A$ *and* $H \in power(B)$ *that* $(\cup_{a\in A} V(n, a)) \cap H = \emptyset$.
- $goals : T \times A \rightarrow 2^G$ *is a function that relates with each pair of sequence number and an agent, the set of goals the agent is interested in.*
- $dep : A \rightarrow T \times A \times 2^A \rightarrow 2^{2^G}$ *is a function that expresses from the point of view of an agent* $a \in A$*, the dependence relation between another agent* $b \in A$ *and a set of other agents regarding the goals of agent* b *in a sequence* $t \in T$.

5.1 Example 1

We explain with an example how to use our formalism and model normative situations. In sequence 2 of our running example, while **Child**'s obligation to select profiles is a normative goal, **Child**'s desire to select the parents it prefers is a personal goal. In this case, personal and normative goals coincide:

The goal g_9, to select parents' profile, is both a personal goal and a normative goal, that is, $goals(2, C) = g_9 \cup O(2, C) = g_9$, where $g_9 \in PG_C$: in sequence 2, agent C has the goal and the obligation to select parents' profiles g_9, where *PG* is personal goal.

$G_C = \cup O(n, C) \cup PG_C$, where $G_C \in G$ is the set of normative goals of agent $C \in A$, $n \in N$ is an adoption norm, $O(n, C)$ is the obligation for C to respect norm n resulting in its normative goals, and $PG_C \in G$ are the personal goals of C.

5.2 Example 2

In this paragraph, we explain the notions of positive and negative consequences to a norm violation. A positive consequence is adding a goal to the existing ones whereas a negative consequence forbid the realization of a goal. We further explain with our example. Let's assume that the parent believes that, in sequence 2, the child depends on the virtual agency to hide its information to parents. However, the parent violates its obligation to respect it and looks up the child's information. One possible sanction is that the parent cannot advertise its profile at the agency any longer, which means that this goal is unrealizable. In the case of the violation sanctioned by the removal of the goal g_2, the obligation $O(n_2, P)$ is not possible any longer as agent P cannot advertise its profile at the agency, it cannot depend on the agency to get the child information any longer. Moreover, agent P cannot achieve its personal goal g_1: adopt a child, any longer as g_2 is a normative goal needed for agent P to achieve g_1. And the violations are: $V^-(n_2, P) = g_2$: agent P violating norm n_2 will not be able to achieve goal g_2, advertise its profile, because g_2 is removed.

As a consequence, the parent cannot adopt a child. Another possible sanction is that the parent must make a donation, e.g. pay a fee, in which case a new goal is added to the parent. As a result, until the parent has fulfill this new obligation, it cannot continue the process.

$dep(P) = (2, C, VA) = g_{14}$: agent P believes that in sequence 2, agent C depends on agent *VA* to achieve its goal g_{14}: no child look up. Where the obligations are:

$O(n_1, C) = g_9$: agent C has the obligation to fulfill norm n_1 to achieve goal g_9, select parent profile.

$O(n_2, P) = g_{14}$: agent P has the obligation to fulfill norm n_2 to achieve goal g_{14}, no look up child.

$V^+(n_2, P) = g_{15}$: agent P violating norm n_2 will not be able to achieve goal g_2, advertise its profile, because a new goal g_{15}, make a donation, is added. Until this new goal is achieved, g_2 cannot be achieved.

In the case of the violation sanctioned with the addition of the goal g_{15}, we note that a mechanism is needed to make sure that the new goal is fulfilled before agent P can further proceeds.

6 Related work

Castelfranchi [11] introduces concepts like groups and collectives from social theory in agent theory, both to enrich agent theory and to develop experimental, conceptual and theoretical new instruments for the social sciences. For further work on the use of the concept of conviviality in computer science and multiagent system see [6, 8, 5, 7]. A large body of work on design has been produced, to only cite a few: the AOSE methodology [20], GAIA [12], the PASSY methodology [13].

7 Summary

- To express the temporal aspects of goal-oriented agents' interactions in multi-agent systems, we use sequences of dependence networks.

- To take into account the individual perspectives of agents for the design of convivial multi-agent systems, we model one dependence network for each agent.
- To design interaction mechanisms that ensure conviviality in multi-agent systems, we use norms.

We apply the social viewpoints on multiagent systems to the concept of conviviality. We use goal refinement within dependence networks by adding and removing goals.

We obtain the following results.

1. By introducing a temporal dimension to out models, we can model the dynamic aspects of conviviality, such as Ashby's observation that enforcing conviviality for the majority re-inforces non-conviviality for minority. Moreover, we can model conviviality by allowing the desired conditions for social interaction, e.g. the creation of new dependence networks and change of the existing ones.

Topics for further research are: We can extend the social models (for example with privacy and community identity) to cover a wider range of notions of conviviality. For instance, Polany's notion of empathy, which needs trust, shared commitments and mutual efforts to build up and maintain conviviality will benefit from such extensions. We can use nested modalities representing agent profiles to model such empathy and related notion of conviviality.

Bibliography

[1] *TOOLS Europe 2001: 38th International Conference on Technology of Object-Oriented Languages and Systems, Components for Mobile Computing, Zurich, Switzerland, 12-14 March 2001*. IEEE Computer Society, 2001.

[2] Systems and software engineering - recommended practice for architectural description of software-intensive systems. Technical report, 2007.

[3] Guido Boella, Luigi Sauro, and Leendert W. N. van der Torre. Social viewpoints on multiagent systems. In *AAMAS*, pages 1358–1359, 2004.

[4] Paolo Bresciani, Anna Perini, Paolo Giorgini, Fausto Giunchiglia, and John Mylopoulos. Tropos: An agent-oriented software development methodology. *Autonomous Agents and Multi-Agent Systems*, 8(3):203–236, 2004.

[5] Patrice Caire. A normative multi-agent systems approach to the use of conviviality for digital cities. In Pablo Noriega and Julian Padget, editors, *Proceedings of The International Workshop on Coordination, Organization, Institutions and Norms in Agent Systems (COIN)*, pages 15–26.

[6] Patrice Caire. Conviviality for ambient intelligence. In Patrick Olivier and Christian Kray, editors, *Proceedings of Artificial Societies for Ambient Intelligence, Artificial Intelligence and Simulation of Behaviour (AISB'07)*, pages 14–19, 2007.

[7] Patrice Caire. A critical discussion on the use of the notion of conviviality for digital cities. In *Proceedings of Web Communities 2007*, pages 193–200, 2007.

[8] Patrice Caire. Designing convivial digital cities. In O. Stock A. Nijholt and T. Nishida, editors, *Proceedings of the 6th Workshop on Social Intelligence Design (SID'07)*, pages 25–40, 2007.

[9] Patrice Caire, Serena Villata, Guido Boella, and Leendert van der Torre. Conviviality masks in multiagent systems. In Lin Padgham, David C. Parkes, Jörg Müller, and Simon Parsons, editors, *AAMAS (3)*, pages 1265–1268. IFAAMAS, 2008.

[10] C. Castelfranchi. Modeling social action for AI agents. *Artificial Intelligence*, 103(1-2):157–182, 1998.

[11] C. Castelfranchi. The micro-macro constitution of power. *Protosociology*, 18:208–269, 2003.

[12] Luca Cernuzzi and Franco Zambonelli. Dealing with adaptive multi-agent organizations in the gaia methodology. In Jörg P. Müller and Franco Zambonelli, editors, *AOSE*, volume 3950 of *Lecture Notes in Computer Science*, pages 109–123. Springer, 2005.

[13] Antonio Chella, Massimo Cossentino, Luca Sabatucci, and Valeria Seidita. Agile passi: An agile process for designing agents. *Comput. Syst. Sci. Eng.*, 21(2), 2006.

[14] Thomas Erickson and Wendy A. Kellogg. Social translucence: an approach to designing systems that support social processes. *ACM Trans. Comput.-Hum. Interact.*, 7(1):59–83, 2000.

[15] Eduardo Rodrigues Gomes, Elisa Boff, and Rosa Maria Vicari. Social, affective and pedagogical agents for the recommendation of student tutors. In *Proceedings of Intelligent Tutoring Systems*, 2004.

[16] Wolfgang Hofkirchner. Unity through diversity.dialectics - systems thinking - semiotics. *Trans, Internet journal for cultural sciences*, 1(15), 2004.

[17] Ivan Illich. *Tools for Conviviality*. Marion Boyars Publishers, August 1974.

[18] Bertrand Meyer. Systematic concurrent object-oriented programming. In Raimund K. Ege, Madhu S. Singh, and Bertrand Meyer, editors, *TOOLS (11)*, page 553. Prentice Hall, 1993.

[19] Bertrand Meyer. At the edge of design by contract. In *TOOLS (38)* [1], page 3.

[20] James Odell, Paolo Giorgini, and Jörg P. Müller, editors. *Agent-Oriented Software Engineering V, 5th International Workshop, AOSE 2004, New York, NY, USA, July 19, 2004, Revised Selected Papers*, volume 3382 of *Lecture Notes in Computer Science*. Springer, 2004.

[21] Rohit Parikh. Social software. *Synthese*, 132(3):187–211, 2002.

[22] Loris Penserini, Anna Perini, Angelo Susi, and John Mylopoulos. High variability design for software agents: Extending tropos. *TAAS*, 2(4), 2007.

[23] M. David Sadek, Philippe Bretier, and E. Panaget. ARTIMIS: Natural dialogue meets rational agency. In *International Joint Conferences on Artificial Intelligence (2)*, pages 1030–1035, 1997.

[24] Jaime Simão Sichman and Rosaria Conte. Multi-agent dependence by dependence graphs. In *Procs. of The First International Joint Conference on Autonomous Agents & Multiagent Systems, AAMAS 2002*, pages 483–490. ACM, 2002.

[25] Claus Weyrich. Orientations for workprogramme 2000 and beyond. Information society technologies report, Information Society Technologies Advisory Group, September, 17 1999.

A Framework for Normative MultiAgent Organisations

Olivier Boissier and Jomi Fred Hübner*

SMA/G2I/ENSM.SE, 158 Cours Fauriel
42023 Saint-Etienne Cedex, France
FirstName.Name@emse.fr

Abstract. The social and organisational aspects of agency have led to a good amount of theoretical work in terms of formal models and theories. From these different works normative multiagent systems and multiagent organisations are particularily considered in this paper. Embodying such models and theories in the conception and engineering of proper infrastructures that achieve requirements of openness and adaptation, is still an open issue. In this direction, this paper presents and discusses a framework for normative multiagent organisations. Based on the Agents and Artifacts meta-model (A&A), it introduces organisational artifacts as first class entities to instrument the normative organisation for supporting agents activities within it.

Keywords: normative system, organisation, artifacts, norm enforcement

1 Introduction

These last years, the global landscape of multiagent technology has pointed out the concepts of norms and organisations for the modeling and programming of such systems [1] [26]. On one side, the introduction of norms have led to the notion of *normative multiagent system*. In [1], it is defined as "a multiagent system organized by means of mechanisms to represent, communicate, distribute, detect, create, modify, and enforce norms, and mechanisms to deliberate about norms and detect norm violation and fulfilment." On the other side, the increasing importance of organisations has promoted an *organisation oriented view* of the programming of MAS [2].

In this paper, we present a framework for *normative multiagent organisations*. Such an approach takes place at the intersection of the normative and the organisation approaches. In this framework, norms are anchored and considered in the context of the organisation of the system. Norms do not refer directly to agents but to primitives related to an organisation such as roles, groups, etc. The set of mechanisms cited above in the definition of normative systems, are

* Supported by French ANR Project ForTrust ANR-06-SETI-006.

[1] The series of COIN (Coordination Organisation Institution and Norms in agent systems) started in 2005 is an example of such an importance.

Dagstuhl Seminar Proceedings 09121
Normative Multi-Agent Systems
http://drops.dagstuhl.de/opus/volltexte/2009/1903

naturally enriched with functions related to the management of the organisation, to the support of the agents in their coordination and participation to the organisation.

As shown in [2], current software engineering approaches on the definition of these systems have led to a general architecture, kind of organisational middleware, composed of services and agents responsible for executing these mechanisms. From an architectural and software point of view, this middleware is generally introduced between the application agents and the agent communication platform. In those cases, the application agents do not have the possibility to take part in the management of the normative organisation to which they participate. As noticed in [20], the agents have become, in some sense, under the 'control' of the organisational middleware with respect to the management and use of *their* organisation. Our motivation in this work consists in the softening of the management of openness promoted by these organisational middlewares. We propose the implementation of the mechanisms supporting the normative organisation at the agents application level with first class entities.

The paper is structured as follows. The next section presents the main foundations that drive and structure our approach for the definition of the framework for normative multiagent organisation. In the following sections, we detail two components of this framework, starting by the *Organisation Modeling language* (cf. Sec. 3). The description of the different *Organisational Artifacts* that support the deployed normative multiagent organisation is splitted in those that are involved in the management and coordination of the organisation (cf. Sec. 4) and in those that support the management and regulation of the normative dimension of the organisation (cf. Sec. 5). The last component of the framework is composed of the *organisation-awareness mechanisms* that can be embedded in the agents of the systems to properly behave in such a framework. This latter component being still under development, some elements will be described instead of an exhaustive description. Before concluding, we provide some discussions and comparisons with the current state of the art. We position more particularily our approach with respect to different challenges presented in [1].

2 General view and foundations

In this section we present the foundational guidelines that have been used for the definition of the framework for the management of normative multiagent organisations. In the sequel, to alleviate the expression, we will use "organisation" instead of "normative organisation".

2.1 Different levels of representation of an organisation

A multiagent organisation can be considered and represented at three different levels: *(i)* the *organisation specification* stating the abstract structure and functioning of the MAS that is independent of the concrete agents that are participating to it, *(ii)* the *organisation entity* built by the different agents in interaction within the organisation according to their autonomous interpretation and obedience of the specified organisation and *(iii)* the *internal organisation entities*, i.e. the local and individual representations of the organisation entity in every agent of the MAS. Let's notice that these levels are not independant. The organisation entity is updated and modified by runtime events related to agents entering and/or leaving the organisation, to group creation, to role adoption, to goal commitment, etc. The global representation of this organisation entity may be not accessible to the agents. It may be only represented in the eyes of an external observer. On the contrary a set of local, potentially inconsistent representations of it may be built and managed by each agent of the organisation. Agents may also be able to decide from these local representations, to adapt and to change the organisation in a bottom-up process, installing a new organisation specification.

To explicitly represent the organisation that is manipulated at these three levels, the framework is composed of an *Organisation Modeling Language* (OML). It is complemented by an *organisation implementation architecture* composed of the set of mechanisms to manage the organisation entity. This architecture is further divided into an organisation infrastructure part and into an agent part.

The *Organisation Modeling Language* (e.g. $\mathcal{M}$OISE$^+$ [24], ISLANDER [12]) is used to express the specification of the multiagent organisation in terms of norms, specific constraints and cooperation patterns that the designer (or the agents themselves) aim at imposing on the agents of the system. Several dimensions are considered in the current litterature: structural, functional, dialogic, etc [8]. One important feature of these OMLs is that norms and constraints do not refer directly to agents but to primitives related to an organisation such as roles, groups, etc. For instance, it can be specified that every agent that adopts a role "student" in a group "laboratory" is obliged to write a thesis. This language defines the explicit representation of the organisation at the three levels described above. Using these representations, the agents can reason on the organisation specification and on their local and individual representation of the organisation entity.

In the litterature, the development of the organisation implementation architecture normally considers both an agent-centred and an organisation-centred point of view [2]. The *agent-centered view* focuses on the organisational agent-level deliberative mechanisms to interpret and reason on the organisation specification and on the organisation entity to which the agents participate [4, 6]. Equiped with such *organisation-awareness mechanisms* agents become *organisation-aware agents*. Let's note that in the sequel, when we will use the term "agent" it is implicitly considered that the agent is an organisation-aware agent. The *organisation-centered* view is mainly concerned with the definition of what we call *organisational infrastructure* (OI) to support, interpret and manage the organisation entity derived from the enactment by the agents of the organisation

[2] In [35] these points of view are called agent and institutional perspectives.

specified with the OML. Thus, the OI provides the agents with global and shared mechanisms related to their participation to the organisation entity.

2.2 Regimented Norms vs Enforced Norms

As stated in the Sec. 1 and [10], a normative multiagent organisation serves as an instrument to control the autonomy of the agents. Its success depends on how the behavioural constraints stated in its specification are ensured in the system. These behavioural constraints are established by the norms that are stated by the specification of the organisation. In the context of this work, a norm can be an obligation, a permission, or an interdiction to perform some action or to achieve some goal. The actions and goals that are considered are related to the problem to solve (e.g. changing the state of some resource) but also related to the management of the organisation itself (e.g. adopting a role, entering in a group). A norm also has a condition that states when it is active and a deadline to be fulfilled [3]. These norms are considered and interpreted in the context of the current organisation entity. Two types of mechanisms can be considered for instrumenting them in the organisation [4]: *regimentation* and *enforcement*.

Regimentation is a mechanism that simply prevents the agents to perform actions that are stated as forbidden by a norm. More precisely, some actions are regimented in order to preserve important features (e.g. wellformedness) of the organisation. For instance, if a group can have at most one agent playing a given role, the organisational action 'adopt this role' in this group is regimented in order to ensure that this constraint is strictly respected. Since this mechanism has to work in an open system, for any kind of agents, it is implemented 'outside' the agents in the organisation infrastructure. Therefore, action regimentation implies the requirement to instrument the MAS with mechanisms preventing the execution of the concerned set of actions and to install them under the strict control of the OI.

Enforcement is a mechanism which is applied after the detection of the violation of some norm. While regimentation is a preventive mechanism, enforcement is a reactive one. From the local point of view of an agent, a norm may be decided to be obeyed or not. From the global point of view of the organisation, the fulfilment/unfulfilment of the norms should be detected, evaluated as a violation or not, and then judged as worth of sanction/reward or not. While detection can be implemented as an automatic process that does not require decision, the evaluation and the judgement need deliberation and reasoning.

[3] We are aware that the concept of norm is broader and more complex than the one used in this paper (e.g. [34] and the Deontic Logic in Computer Science workshop series [18]). For the present paper however this simple and informal definition is enough to discuss the proposal.

[4] This classification is based on the proposal described in [19, 15]. However, we present them in a more specific context: regimentation is applied only to preventing the execution of organisational actions and enforcement is applied for the the other cases.

Instrumenting norms of the organisation either as regimentations or enforcement mechanisms depends on which side the designer wants to give more weight. Looking further at the functions used in the corresponding mechanisms, two classes can be indeed distinguished: *(i)* management of regimentation in terms of interpretation of the considered norms and checking their satisfaction before executing changes in the organisation entity, *(ii)* management of enforcement in terms of status detection, evaluation of this status and judgement on violation or not followed by sanction execution.

2.3 First class entities for the Management of Normative Organisation

As mentioned in the introduction, the engineering of organisation infrastructure in the litterature has led to the proposals of organisation middleware installing the OI as a separate layer that cannot be managed by the agents participating to the application which is developed on top of this middleware. However, as argued in [20], even if the OI aims at supporting and controling the agents in their participation to the organisation entity, we consider that it should also be managed by those agents.
To solve this problem, we propose to design and develop it *within* the multiagent layer where the application is developed with the first class abstractions that are used to develop it. The choice of these first class abstractions must be considered with care, since, as stated in the previous section, the management of norms in the OI strongly depends on a regimentation or an enforcement view.
Basing our approach on the basic A&A (Agents and Artifacts) meta-model presented in [33], the organisation infrastructure of the framework, called ORA4MAS [25], proposes a set of artifacts, called *organisational artifacts*. The agents can use these artifacts to instrument *(i)* the multiagent environment which is no more a merely passive source of agent perceptions and target of agent actions and *(ii)* the organisation entities living upon in order to interpret and manage them according to the way they are specified with the OML. Given the deliberative nature of some of the mechanisms involved in the norm enforcement (evaluation, judgement and sanction), the overall picture of ORA4MAS accounts also for *organisational agents* (cf. Fig. 1).
We use here the adjective "organisational" to identify those agents and artifacts of the MAS which are part of the OI. They are responsible for activities and encapsulate functionalities concerning the management and enactment of the organisation. It is however possible, depending on the application requirements, that agents participating also to the proper solving and functioning of the application endorse the "role" of organisational agents.

Analogously to the human case, *organisational artifacts* are used here to reify and modularise the functional-part of the organisation management machinery. As in the A&A model, they are non-autonomous function-oriented entities, designed to provide resources and tools that agents can create and use. They are focused on the organisation entity management activities. As the cognitive

artifacts proposed in the A&A model, they constitute a distributed set of organisational resources and tools that can be perceived and used by agents as first-class entities. They can be dynamically adapted and possibly replaced (by agents themselves) during the organisation lifetime. As cognitive artifacts, the organisational artifact function is partitioned in a set of operations, which agents can trigger by acting on artifact usage interface. The usage interface provides all the controls that make it possible for an agent to interact with an organisational artifact, that is to use and observe it. Agents can use an organisational artifact by triggering the execution of operations through the usage interface and by perceiving observable events generated by the artifact itself, as a result of operation execution and evolution of its state. Besides the controls for triggering the execution of operation, an organisational artifact can have some observable properties, i.e. properties whose value is made observable to agents, without necessarily executing operations on it. Organsational artifacts then mediate the access of agents to organisation resources and support participation of these agents to organisation activities. For instance, to adopt a role an agent has to use the appropriate artifact.
Considering the normative dimension of the organisation, organisational artifacts encapsulate also organisational norms and functionalities, such as enabling, mediating, and ruling agent interaction, tracing and ruling resource access, and so on. Regimentations of norms (see Fig. 1) are implemented in the organisational artifacts. For instance, let's consider the case of a regimented adoption of role. The operation will be successfully executed only in the case the agent is allowed to adopt the role, otherwise the adoption fails. Since it is possible to link organisational artifacts with cognitive artifacts that mediate the access of agents to resources, it is thus possible to imagine to regiment also the access to those resources by the way of the organisational artifact. In the case of enforcement of norms, the functionality provided by the artifacts consists in the *detection* and showing (by means of observable properties) the non fulfilment of a norm. We consider that agents (organisational ones or not depending on the application) should be informed of current status of the norm and can evaluate the existence of violation or not and take the better decision regarding the application objectives.

The *organisational agents* embed dedicated reasoning and strategies related to the management of the organisation. They can be dedicated agents or agents of the application having special knowledge. They dynamically articulate, manage, regulate and adapt the organisation entity by creating, linking and manipulating the organisational artifacts, which are discovered and used by the agents to work inside the organisation entity, according to the specified organisations. Such activities typically include observing artifacts dynamics and possibly intervening, by changing and adapting artifacts or interacting directly with other agents, so as to improve the overall (or specific) organisation processes or taking some kinds of decisions when detecting violations. As an example, in the context of the $\mathcal{M}$OISE$^+$ model, one or multiple *scheme manager* agents can be introduced, responsible for monitoring the dynamics of the execution of a scheme by observing a specific artifact. The scheme artifact and scheme manager agents are designed so as that the artifact allows for violation of the deontic rules concerning the commitment of missions by agents playing some specific roles, and then the decision about what action to take – after detecting the violation – can be in charge of the manager agent.

Organisational artifacts and organisational agents create a sort of explicit organisational infrastructure on which the organisation entity is deployed, revealed to the agents as available tools in the environment. Regimentation and detection mechanisms into artifacts, whereas evaluation and judgement mechanisms involved in inforcement are implemented into the agents. ORA4MAS is thus able to ensure that important properties of the organisation entity hold while agents keep their autonomy with respect to the constraints considered as norms to enforce.

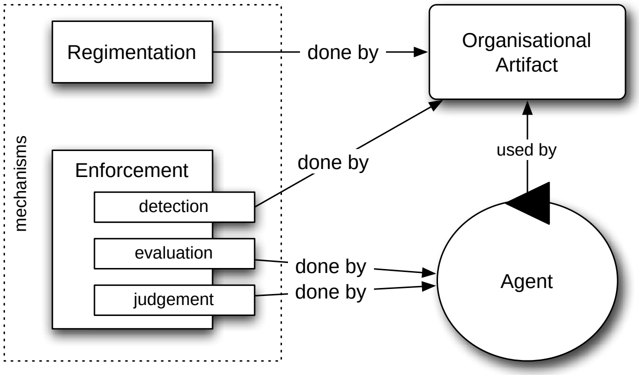


Fig. 1. General relation between the mechanisms that implement the norms and the organisational agents and artifacts of ORA4MAS

Given the sketching of the foundational guidelines underlying our framework, we describe in the next section its different components and how the full-fledged $\mathcal{M}$OISE$^+$ organisational model [5] can be implemented with organisational artifacts. Detailed examples of the use of the framework can be found in [20] and in [22]. ORA4MAS is realised on top of CARTAGO infrastructure [32], embedding algorithms used in $\mathcal{S}$-$\mathcal{M}$OISE$^+$ [23]. CARTAGO is integrated with ***Jason*** [3], 2APL [9], and JADEX [29] — these integrations are presented in [31, 28].

In this paper, descriptions of the organisation-awareness mechanisms and of the organisational agents are not given. Even if some work has been realized in the different examples that we have developed, we don't have yet generic

[5] Different OMLs require a different set of suitable artifacts and agents. For instance, in the AGR organisational model [14], we can conceive artifacts to manage groups; for ISLANDER [12], the artifacts can be used to manage the scenes.

architectures of such agents. Examples of organisation-awereness mechanisms may be found for instance in [5].

3 Normative Organisation Modeling Language

The current version of the framework uses the $\mathcal{M}$OISE$^+$ OML[24] as the language to define and describe explicitly an organisation. This language decomposes the specification of an organisation into three independent dimensions: structural, functional, and deontic dimensions [6]. The structural dimension focuses on the specification of the *roles*, *groups*, and *links* of the organisation. The definition of roles states that when an agent decides to play some role in a group, it is accepting some behavioural constraints related to this role. The functional dimension specifies how the *global collective goals* should be achieved, i.e. how these goals are decomposed into global *plans*, grouped into coherent sets (called *missions*) to be allocated to roles. The decomposition of global goals results in a goal-tree, called *scheme*, where the leaves-goals can be achieved individually by agents. The deontic dimension glues the structural dimension with the functional one by the specification of the roles' *permissions* and *obligations* for missions.

The detailed syntax and definition of the language is described in [21]. Let's stress that, agents and the OI interpret that declarative organisation specification. This language is founded on components represented by predicates and functions. Considering an organisation specification, $\mathcal{G}$, $\mathcal{R}$, $\mathcal{S}$, $\mathcal{M}$, Φ denote respectively the set of all group specifications, the set of all roles, the set of all scheme specifications, the set of all missions, and the set of all goals. We present here only some predicates that are used in the sequel of the paper:

- $compat(g, \rho, C)$: is true iff the role ρ ($\rho \in \mathcal{R}$) is *compatible* with all roles in the set C ($C \subseteq \mathcal{R}$) when played in the group g ($g \in \mathcal{G}$) (two roles are compatible if they can be adopted by the same agent);
- $mission\_scheme(m, s)$: is true iff the mission m ($m \in \mathcal{M}$) *belongs to* the scheme s ($s \in \mathcal{S}$);
- $goal\_mission(\varphi, m)$: is true iff the goal φ ($\varphi \in \Phi$) *belongs to* the mission m ($m \in \mathcal{M}$);
- $obl(\rho, m)$: is true iff the role ρ has an *obligation relation* to the mission m;
- $per(\rho, m)$: is true iff the role ρ has a *permission relation* to the mission m;
- $goal\_role(\varphi, \rho)$: is true iff goal φ is part of one of the obliged missions of role ρ;

Similarly, functions of this language that are considered in the sequel are:

[6] Extensions are currently on the way to integrate an extended version of it in the framework. These extensions have been developped in the Moise-Inst OML [17]. This OML proposes an enriched deontic dimension with more expressive normative expressions and also a supplementary dimension, called context specification, stating the a priori evolution of the organisation.

- $maxrp : \mathcal{R} \times \mathcal{G} \rightarrow \mathbb{Z}$: returns the maximum number of players of a role in a group, i.e. upper bound of the *role cardinality*;
- $minrp : \mathcal{R} \times \mathcal{G} \rightarrow \mathbb{Z}$: returns the minimum number of players of a role within a group, necessary for that group to be considered *well-formed* (i.e. lower bound of the *role cardinality*);
- $maxmp : \mathcal{M} \times \mathcal{S} \rightarrow \mathbb{Z}$: returns the maximum number of agents that can commit to a mission in a scheme (i.e. upper bound of the *mission cardinality*);
- $minmp : \mathcal{M} \times \mathcal{S} \rightarrow \mathbb{Z}$: returns the minimum number of agents that have to commit to a mission within a scheme for that scheme to be considered well-formed regarding that mission (i.e. lower bound of the *mission cardinality*).

4 Organisational Artifacts for Organisation Coordination

Derived from the OML presented in previous section, we describe here and in Sec. 5 the basic set of artifacts of ORA4MAS [25] that constitutes the building blocks for the support of the 'reification' of the structural specification (SS), functional specification (FS), and deontic specification (DS) of $\mathcal{M}$OISE$^+$. We focus here on the management of the organisation. In Sec. 5, we will present organisational artifacts in relation to the normative content of the organisation entity.
The basic set of organisational artifacts considered here accounts for: OrgBoard, GroupBoard artifacts, SchemeBoard artifacts. The instrumentation of the organisational entity with those artifacts is done as follows: one and only one OrgBoard is used to keep track of the current state of the deployed organisational artifacts supporting the current organisational entity in the overall, one GroupBoard for each instance of group of agents used to manage the life-cycle of this specific instance of a group, one SchemeBoard for each social scheme being executed by the agents used to support and manage the execution of it.

The organisational artifacts are linked together to allow the synchronisation of some their operations and to share information required to maintain a coherent and consistent state of the organisation entity. The OrgBoard is linked to all the other organisational artifacts of the organisation entity. The GroupBoard is linked to all SchemeBoard that manage schemes involving for their execution agents that are member of the corresponding group. Each SchemeBoard is linked to exactly one NormativeBoard (see next section) that verifies the status of the norms related to the execution of the scheme.

In the following we briefly describe these artifacts. We consider just a core set of the characteristics and functioning of the artifacts, skipping most details that would make heavy the overall understanding of the approach.

4.1 OrgBoard artifact

An abstract representation of the OrgBoard is depicted in Fig. 2. The observable properties of this organisational artifact are:

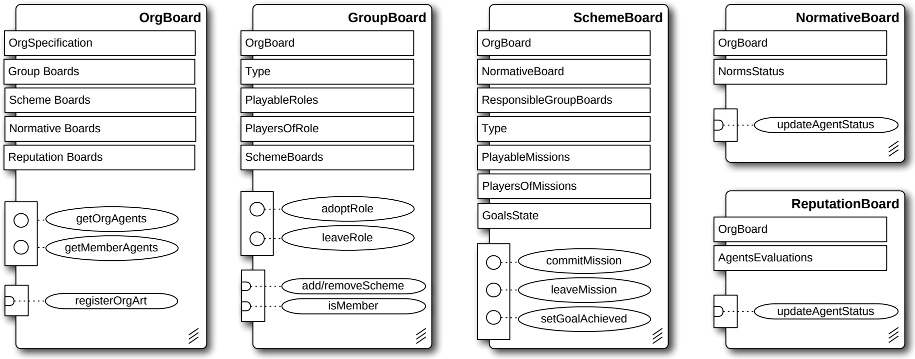


Fig. 2. Basic kinds of artifacts in ORA4MAS, with their usage interface, including operations (represented by circles) and observable properties (represented by rectangles with circles in their center), and the link interface (represented by rectangles with circles on their left side)

- OrgSpecification: specification of the organisation entity written in the $\mathcal{M}$OISE$^+$ OML. Agents may use this observable property to get the organisational specification. They can then reason about it and decide whether they want or no to enter in the organisation.
- GroupBoards, SchemeBoards, and NormativeBoards and ReputationBoards: identifiers of all instances of GroupBoard, SchemeBoard, NormativeBoard, ReputationBoard, respectively, within the organisation entity. Generally speaking, these observable properties make it possible for agents observing an OrgBoard to know the current set of organisational artifacts instrumenting the organisation entity.

The usage interface of the OrgBoard has the following operations:

- getOrgAgents(): used to get the set of agents having the status of organisational agent in the organisation entity (this status is set during the deployment of the system).
- getMemberAgents(): used to get the set of all agents playing at least one role within a group of the organisation entity [7]

The main link operation of the OrgBoard is registerOrgArt. It is used in the initialisation process of each new instance of GroupBoard, SchemeBoard, and NormativeBoard to be registered as an artifact linked to the organisation entity represented by the OrgBoard.

[7] This operation would be implemented as an observable properties; however due to distributed characteristic of the information (they are managed by all GroupBoard), maintaining an uptodate observable property would be time consuming. Using an operation, the list of member agents is a cache, computed only on demand.

4.2 GroupBoard artifact

A GroupBoard is an organisational artifact providing functionalities to manage a group. Each GroupBoard is attached to a group in the organisation entity instanciated from a specific group specification. It maintains a consistent state of that group by regimenting some norms stating essential structural properties. For instance, whenever some agent asks for a role adoption in the group managed by the GroupBoard, the GroupBoard regiments a set of norms that state when a role can be adopted: (1) the role belongs to its group specification; (2) each role that the agent already plays is specified as compatible with the new role; and (3) the number of players is lesser or equals than the maximum number of players defined in the group's compositional specification.

As an artifact, the GroupBoard has some observable properties (Fig. 2) that enable agents to know which are the available roles and their constraints, which are the participant agents, and which are the other organisational artifacts linked to the GroupBoard. Among them, the most relevant are:

- OrgBoard: is the reference to the OrgBoard that represents the organisation entity to which the group belongs.
- Type: is the identification of the group specification in the structural specification (an element of $\mathcal{G}$).
- PlayableRoles: contains all roles that can still be adopted in this group, i.e. those which the number of players is not the maximum yet. This property changes whenever a new agent enters into the group by adopting a role.
- PlayersOfRole: contains the names of all agents belonging to the group and their corresponding roles.
- SchemeBoards: contains a set of all schemes the group is responsible for.

The usage interface accounts for the following operations:

- adoptRole(ρ): used by an agent to adopt a new role in the group, where $\rho \in \mathcal{R}$ is the identifier for a role in the Structural Specification.
- leaveRole(ρ): used by an agent to give up the role ρ that it had adopted previously.

The link operations of a GroupBoard manage the coordinations with its linked organisational artifacts. Among them, we have:

- addSchemeBoard(sb): used by a SchemeBoard initialisation process to notify the GroupBoard that it is responsible for the scheme sb. The GroupBoard updates accordingly its SchemeBoards observable property.
- removeSchemeBoard(sb): used by a SchemeBoard linked to the GroupBoard, to notify that the GroupBoard is no more responsible for the execution of the scheme sb. The GroupBoard updates accordingly its SchemeBoards observable property.
- isMember(α): used by a SchemeBoard to request whether an agent α is member (i.e. is playing at least one role) of the group managed by the GroupBoard.

Before presenting the norms that a GroupBoard regiments, let's introduce some predicates related to the internal state of this artifact. Let $\mathcal{GB}$, $\mathcal{R}_g$, and $\mathcal{A}$ be respectively the set of current group boards, the set all roles that can be played in a group board created from specification g, and the set of all agents participating to the organisation entity.

- $group\_type(gb, g)$: is true iff the group board $gb \in \mathcal{GB}$ has been created based on the group specification $g \in \mathcal{G}$ (cf. Sec. 3);
- $plays(\alpha, \rho, gb)$: is true iff agent $\alpha \in \mathcal{A}$ plays role ρ in the group board $gb \in \mathcal{GB}$;

The function *rplayers* returns the number of current players of the role ρ in the group gb.

$$\begin{aligned} rplayers &: \mathcal{R} \times \mathcal{GB} \rightarrow \mathbb{Z} \\ rplayers(\rho, gb) &\overset{\mathsf{def}}{=} |\{\alpha \mid plays(\alpha, \rho, gb)\}| \end{aligned} \tag{1}$$

Given the above definitions and the functions *maxrp* and *minrp* (cf. Sec. 3), we are able to define the *wellformedness property* of a group

$$\begin{aligned} well\_formed(gb) \leftarrow\ & group\_type(gb, g)\ \wedge \\ & \forall_{\rho \in \mathcal{R}_g}\ rplayers(\rho, gb) \geq minrp(\rho, g)\ \wedge \\ & \qquad rplayers(\rho, gb) \leq maxrp(\rho, g) \end{aligned} \tag{2}$$

Since role adoption is the very action that may bring a group in an inconsistent state, two norms bearing on this organisational action are regimented by a GroupBoard: *role compatibility* norm and *role cardinality* norm. In the following norms are represented as a pair. The first argument is the condition part stating when the norm is active. The second argument is the action part stating the obligation, permission, or interdiction.

Role compatibility norm: In the $\mathcal{M}$OISE$^+$ language, roles are incompatible unless explicitly stated the contrary in the organisation specification. When two roles ρ_1 and ρ_2 are specified as compatible inside a group g ($compat(g, \rho_1, \{\rho_2\})$), it implies that an agent that plays ρ_1 in a group board gb created from the specification g cannot perform the operation $\mathsf{adoptRole}(\rho_i)$ for any $i \neq 2$ on the corresponding artifact. This constraint on role adoption is formalised by the following norm:

$$\begin{aligned} &(plays(\alpha, \rho, gb)\ \wedge\ group\_type(gb, g)\ \wedge\ compat(g, \rho, C), \\ &\forall_{\rho_i \in \mathcal{R} \setminus C}\ FORBIDDEN_\alpha\ \mathsf{adoptRole}(\rho_i)) \end{aligned} \tag{3}$$

The condition of the norm (the first line) is a conjunction of predicates. Its evaluation is given by the particular status of the group board (that defines whether $plays(\alpha, \rho, gb)$ and $group\_type(gb, g)$ hold or not) and by the structural specification used by the artifact (that defines whether $compat(g, \rho, C)$ holds or not). The action part of the norm (the second line) states that it is forbidden for agent α to execute the action adoptRole on any role that does not belong to the set of compatible roles C. Based on this norm, as soon as an agent adopts a role (activating the norm), the adoption of other roles that are not explicitly stated compatible are forbidden for it.

Role cardinality norm: The number of players of a role in a group is limited by the function $maxrp(\rho, g)$ defined from the Structural Specification. The following norm constrains the role adoption based on the cardinality of the role:

$$\begin{aligned}&(group\_type(gb, g) \ \wedge \ rplayers(\rho, gb) \geq maxrp(\rho, g), \\ &\forall_{\alpha \in \mathcal{A}} \forall_{\rho \in \mathcal{R}} \ FORBIDDEN_{\alpha} \ \mathsf{adoptRole}(\rho))\end{aligned} \tag{4}$$

Since these two norms are of the type 'action interdiction', they can be easily implemented in the artifact: whenever the adoptRole operation is triggered by the agent α, the condition of all norms are checked using the structural specification and the current state of the group artifact. If the condition of some of these norms holds, the execution of the corresponding operation is denied.

4.3 SchemeBoard artifact

A SchemeBoard is an organisational artifact providing functionalities to manage the execution of a social scheme. Each SchemeBoard is instantiated upon a specific social scheme specification of the Functional Specification. It coordinates the commitments to missions and the achievement of goals by managing the dependencies between the missions and the goals as described in the social scheme specification. The lifecycle of a SchemeBoard is organised along three phases: formation, goal achievement and finishing. In the formation phase, agents commit to the missions of the scheme. A property of wellformedness conditions the transition to the second phase. A scheme is well-formed if the mission cardinalities are satisfied, i.e. there are enough agents committed to the missions (see below for a more formal definition). In the second phase, goals should be fulfilled by the agents. Each agent is expected to achieve the goals of the missions it is committed to. When the root goal of the scheme is satisfied, the third phase starts and the scheme can be finished and removed from the organisation entity (i.e. the corresponding artifact is destroyed).

During the execution of a scheme, its goals can be in three different states: *waiting*, *possible*, *achieved*. The *waiting* state is the initial state of every goal. In such a state, a goal can not be pursued by the agents. Its change of state depends on the achievement of other goals (called pre-conditions for a goal) or of the wellformedness of the scheme (in case the goal has no pre-conditions). The set of pre-conditions for a goal is deduced from the goal decomposition tree of the scheme. When all pre-conditions of a goal are satisfied and the scheme is well-formed, the state of a goal is changed to *possible*. Then the agent(s) committed to a mission containing that goal can start to achieve it. Let's note that the change from the state *waiting* to *possible* is performed by the SchemeBoard, whereas the change from the state *possible* to *achieved* is performed by the agents.

The observable properties of a SchemeBoard are defined to make an agent able to monitor the overall dynamics concerning execution of the corresponding scheme. It is thus possible for an agent to be aware of which missions are assigned to which agents, which goals are achieved and which can be pursued. Among the observable properties, the most important are the following (Fig. 2):

- OrgBoard: is the reference to the OrgBoard that represents the organisation entity in which the scheme is being executed.
- NormativeBoard: is the reference to the NormativeBoard linked to the scheme [8].
- ResponsibleGroupBoards: contains the references to the GroupBoard that are responsible for the scheme.
- Type: is the identification of the scheme specification in the functional specification (an element of $\mathcal{S}$).
- PlayableMissions: contains all missions that can still be committed to in the scheme.
- PlayersOfMission: contains all the agents committed to a mission of the scheme and their corresponding mission.
- GoalsState: contains the current state of the goals of the scheme.

The usage interface provides the following operations:

- commitMission(m): used by an agent to commit to a mission $m \in \mathcal{M}$;
- leaveMission(m): used by an agent to give up a mission m it is committed to;
- setGoalAchieved(φ): used by an agent to set the state of a goal to *achieved*.

As for the GroupBoard, we define the following predicates bearing on the current state of a SchemeBoard. Let $\mathcal{SB}$ and $\mathcal{M}_s$ be respectively the set of current scheme boards and the set all missions that can be played in a scheme board created from specification s.

- $scheme\_type(sb, s)$: is true iff the scheme board $sb \in \mathcal{SB}$ is created based on the scheme specification $s \in \mathcal{S}$ (the type of a scheme board is defined in its creation);
- $resp\_group(gb, sb)$: is true iff the group $gb \in \mathcal{GB}$ is responsible for the execution of the scheme $sb \in \mathcal{SB}$;
- $committed(\alpha, m, sb)$ is true iff the agent $\alpha \in \mathcal{A}$ is committed to the mission $m \in \mathcal{M}$ in the scheme $sb \in \mathcal{SB}$;
- $achieved(\varphi, sb)$: is true iff the goal φ is already achieved in the scheme sb;
- $possible(\varphi, sb)$: is true iff the state of the goal φ is possible in the scheme sb; considering Φ' the set of all goals that are pre-condition of φ, this predicate can be deduced by

$$possible(\varphi, sb) \leftarrow \bigwedge_{\varphi' \in \Phi'} achieved(\varphi', sb) \ \wedge \ well\_formed(sb) \tag{5}$$

- $succeeded(s)$ it is true that the scheme s has finished successfully.

The function $mplayers$ returns the number of current players of the mission m in the scheme of sb.

$$\begin{aligned} &mplayers : \mathcal{M} \times \mathcal{SB} \rightarrow \mathbb{Z} \\ &mplayers(m, sb) \overset{\texttt{def}}{=} |\{\alpha \mid committed(\alpha, m, sb)\}| \end{aligned} \tag{6}$$

[8] This observable property indirectly links all responsible groups of the scheme to the normative board of the same scheme.

Given the above definitions and the functions *maxmp* and *minmp* (cf. Sec. 3), we are able to define the *wellformedness property* of a scheme:

$$\begin{aligned} well\_formed(sb) \leftarrow\ & scheme\_type(sb, s) \wedge \\ & \forall_{m \in \mathcal{M}_s}\ mplayers(m, sb) \geq minmp(m, s) \wedge \\ & \qquad mplayers(m, sb) \leq maxmp(m, s) \end{aligned} \tag{7}$$

Mission commitment norm: Analogously to the *role cardinality* norm, we define a *mission commitment* norm to forbid an agent to commit to missions in a scheme. The number of agents already committed to a mission constrains the action commitMission (mission cardinality). Another constraint is that only agents that play some role in a responsible group for the scheme can commit to a mission in the scheme. We define the mission commitment norm as follows:

$$\begin{aligned} &((scheme\_type(sb, s)\ \wedge\ mplayers(m, sb) \geq maxmp(m, s)) \vee \\ &(resp\_group(gb, sb)\ \wedge\ \neg plays(\alpha, \rho, gb)), \\ &FORBIDDEN_\alpha\ \mathsf{commitMission}(m)) \end{aligned} \tag{8}$$

The implementation of this norm follows the same algorithm used by the GroupBoard: whenever an agent attempts to commit to a mission, if the condition of the norm holds, the operation is denied.

Note that in the current version of ORA4MAS, there is no regimentation on the leaving of a mission or of a role. We consider that these organisational actions should give rise to enforcement and violation. For instance, we could imagine to detect a violation when a mission or a role are left while still having goals of the mission to be achieved. Following the detection of violation, sanctions have to be decided by organisational agents.

5 Organisational Artifacts for Organisation Regulation

Pursuing the description of the basic set of organisational artifacts building ORA4MAS, we turn to the organisational artifacts in relation with the normative dimension of the organisation which is connected to the enforcement mechanisms and to the regulation of the organisational entity. In the current state of the framework, two kinds of such organisational artifacts have been defined: NormativeBoard and ReputationBoard artifacts. They are used to maintain and provide information concerning the agents compliance or not to norms. These artifacts don't provide any operation to the agents since their function is to detect and show as observable properties information related to the current status of the norms given the agents' behaviour related to the groups and scheme they are linked to.

5.1 NormativeBoard artifact

The NormativeBoard artifact (Fig. 2) embeds the functionalities to manage the specification concerning permissions and obligations defined between roles of the

Structural Specification and missions of the Functional Specification. There is one link operation (updateAgentStatus) used by the assigned scheme and groups to trigger an update of the current status concerning a particular agent whenever this agent has performed some operation in the scheme or group.

The norms that are considered in this artifact are not implemented by regimentation, since we would like to allow the agents to violate them. Their implementation is thus not as simple as the implementation of the norms of group and scheme artifacts (where only interdictions are considered and regimentation is used as the mechanism). The NormativeBoard manages the state of the norms as follows (more details are available in [20]). The state of a norm is initially *inactive*. It becomes *active* when its condition holds. When the agent executes the action as it is stated in the action part of the norm, the status of the norm becomes *fulfilled*. In the other case, i.e. the agent does not behave in time accordingly to the action part of the norm, the status of the norm becomes *unfulfilled*.

The set of norms are defined from the deontic specification and the current state of related artifacts. As examples, in the sequel some of these norms are presented.

Obligation to commit to a mission: Based on the deontic relations $obl(\rho, m)$ included in the organisation specification (as defined in section 3), the roles played by the agents (as defined in the section 4.2), and the current number of agents committed to a mission (an agent is not obliged to commit to a mission if the minimum number of players is already achieved), the following norm is defined:

$$\begin{aligned}&(obl(\rho, m) \ \wedge \ plays(\alpha, \rho, gb) \ \wedge \ resp\_group(gb, sb) \ \wedge \\ &scheme\_type(sb, s) \ \wedge \ mission\_scheme(m, s) \ \wedge \\ &mplayers(m, sb) < minmp(s), \\ &OBLIGED_\alpha \ \mathsf{commitMission}(m))\end{aligned} \tag{9}$$

The three first lines of this norm are the condition that states when the norm is active and the last line represents the obligation for the target agent.

Permission to commit to a mission: Based on deontic relations $per(\rho, m)$ included in the organisation specification, and the roles played by the agents, the following a norm which is defined as an interdiction as follows:

$$\begin{aligned}&(\neg(per(\rho, m) \ \wedge \ plays(\alpha, \rho, gb) \ \wedge \ resp\_group(gb, sb) \ \wedge \\ &scheme\_type(sb, s) \ \wedge \ mission\_scheme(m, s)), \\ &FORBIDDEN_\alpha \ \mathsf{commitMission}(m))\end{aligned} \tag{10}$$

Obligation to achieve a goal: Once an agent α is committed to a mission m, it is obliged to fulfil the possible goals of the mission. The norm below specifies that rule.

$$\begin{aligned}&(committed(\alpha, m, sb) \ \wedge \ goal\_mission(\varphi, m) \ \wedge \ possible(\varphi, sb), \\ &OBLIGED_\alpha \ \varphi)\end{aligned} \tag{11}$$

5.2 Artifact for Instrumenting Reputation Processes

Inspired by the concept of *reputation artifact* proposed in [7, p. 101], ORA4MAS is enriched with such a type of artifact in order to provide first class constructs which can be easily used to support the reputation processes. It serves as an indirect sanction instrument for norms enforcement. While direct sanctions are applied when the violation is detected, indirect sanctions have long term results, as is the case of reputation.

This artifact is linked to all the organisational artifacts described in the previous section and to the NormativeBoard artifacts. It can be observed by all agents inside the organisation. The other artifacts notify it about the current state of the organisation. This information is used to compute an *evaluation* for each agent member of the organisation entity. This evaluation is published as an observable property of the artifact. It is important to notice that the evaluation is not the reputation of the agent, as remarked in [7], reputation is a *shared voice* circulating in a group of agents. This artifact is indeed an instrument to influence the reputation of the agent.

Several criteria may be used to evaluate an agent inside an organisation. Herein we chose to evaluate an agent in the context of the roles and missions it is concerned by along three criteria: obedience, pro-activeness, and result.

- *obedience* of an agent is computed by the number of obliged goals it achieves. The goals an agent is obliged to achieve are defined by the deontic specification. All obliged goals that have not been achieved until its deadline are considered as a possible violation (this detection is provided by the normative board). Let's define the following functions: general mission obedience function ($o : \mathcal{A} \rightarrow [0,1]$) and obedience in the context of a particular mission function ($o_m : \mathcal{A} \times \mathcal{M} \rightarrow [0,1]$) and obedience in the context of a particular role ($o_r : \mathcal{A} \times \mathcal{R} \rightarrow [0,1]$). They are computed as follows (in the equations # is a function that returns the size of a set):

  $$o(\alpha) = \frac{\#\{\varphi \mid \mathit{obliged}(\alpha,\varphi) \ \wedge \ \mathit{achieved}(\alpha,\varphi)\}}{\#\{\varphi \mid \mathit{obliged}(\alpha,\varphi)\}}$$

  $$o_m(\alpha,m) = \frac{\#\{\varphi \mid \mathit{obliged}(\alpha,\varphi) \ \wedge \ \mathit{goal\_mission}(\varphi,m) \ \wedge \ \mathit{achieved}(\alpha,\varphi)\}}{\#\{\varphi \mid \mathit{obliged}(\alpha,\varphi) \ \wedge \ \mathit{goal\_mission}(\varphi,m)\}}$$

  $$o_r(\alpha,\rho) = \frac{\#\{\varphi \mid \mathit{obliged}(\alpha,\varphi) \ \wedge \ \mathit{goal\_role}(\varphi,\rho) \ \wedge \ \mathit{achieved}(\alpha,\varphi)\}}{\#\{\varphi \mid \mathit{obliged}(\alpha,\varphi) \ \wedge \ \mathit{goal\_role}(\varphi,\rho)\}}$$

  $o(\alpha) = 1$ means that the agent α achieved all its obligation and $o(\alpha) = 0$ means it achieved none. $o_m(\alpha,m) = 1$ means that the agent achieved all goals when committed to the mission m, and $o_r(\alpha,\rho) = 1$ means that the agent achieved all goals when playing the role ρ.
- The *pro-activeness* of an agent is computed by the number of goals an agent achieves such that it is not obliged to fulfil that goal in a scheme. The general pro-activeness function ($p : \mathcal{A} \rightarrow [0,1]$) and the pro-activeness in the context

of a particular mission ($p_m : \mathcal{A} \times \mathcal{M} \to [0,1]$) and role ($p_r : \mathcal{A} \times \mathcal{R} \to [0,1]$) are defined as follows:

$$p(\alpha) = \frac{\#\{\varphi \mid \mathit{achieved}(\alpha,\varphi) \ \wedge \ \neg \mathit{obliged}(\alpha,\varphi)\}}{\#\Phi \, \#\mathcal{S}}$$

$$p_m(\alpha,m) = \frac{\#\{\varphi \mid \mathit{achieved}(\alpha,\varphi) \ \wedge \ \neg \mathit{obliged}(\alpha,\varphi) \ \wedge \ \mathit{goal\_mission}(\varphi,m)\}}{\#\{\varphi \mid \mathit{committed}(\alpha,m,\_) \ \wedge \ \mathit{goal\_mission}(\varphi,m)\}}$$

$$p_r(\alpha,\rho) = \frac{\#\{\varphi \mid \mathit{achieved}(\alpha,\varphi) \ \wedge \ \neg \mathit{obliged}(\alpha,\varphi) \ \wedge \ \mathit{goal\_role}(\varphi,r)\}}{\#\{\varphi \mid \mathit{committed}(\alpha,m,\_) \ \wedge \ \mathit{goal\_mission}(\varphi,m) \ \wedge \ \mathit{goal\_role}(\varphi,r)\}}$$

$p(\alpha) = 1$ means that the agent achieved all goals it is not obliged to (a highly pro-active behaviour) and $p(\alpha) = 0$ means the contrary.

- The *results* of an agent is computed by the number of successful execution of scheme where it participates. It does not depend on the achievement of the goals in the scheme. It means the agent somehow share the success of the scheme execution and likely has helped for the success. The general results function ($r : \mathcal{A} \to [0,1]$) and the results in the context of a particular mission ($r_m : \mathcal{A} \times \mathcal{M} \to [0,1]$) and role ($r_r : \mathcal{A} \times \mathcal{R} \to [0,1]$) are defined as follows:

$$r(\alpha) = \frac{\#\{s \mid \mathit{committed}(\alpha,\_,s) \ \wedge \ \mathit{succeeded}(s)\}}{\#\{s \mid \mathit{committed}(\alpha,\_,s)\}}$$

$$r_m(\alpha,m) = \frac{\#\{s \mid \mathit{committed}(\alpha,m,s) \ \wedge \ \mathit{succeeded}(s)\}}{\#\{s \mid \mathit{committed}(\alpha,m,s)\}}$$

$$r_r(\alpha,\rho) = \frac{\#\{s \mid \mathit{committed}(\alpha,m,s) \ \wedge \ \mathit{succeeded}(s) \ \wedge \ \mathit{obl}(\rho,m)\}}{\#\{s \mid \mathit{committed}(\alpha,m,s) \ \wedge \ \mathit{obl}(\rho,m)\}}$$

$r(\alpha) = 1$ means that all schemes the agent participated have finished successfully and $r(\alpha) = 0$ means the contrary.

Unlike the previous two criteria, the results value of an agent cannot be increased by the agent itself. This evaluation depends on the performance of all agents committed to the same scheme, creating thus a dependence among them. The selection of good partners is therefore important and the reputation artifact could be used for that purpose.

The aforementioned criteria are combined into a single overall evaluation of an agent ($e : \mathcal{A} \to [0,1]$) by the following weighted mean:

$$e(\alpha) = \frac{\gamma \, o(\alpha) + \delta \, p(\alpha) + \epsilon \, r(\alpha)}{\gamma \ + \ \delta \ + \ \epsilon}$$

$$e_m(\alpha,m) = \frac{\gamma \, o(\alpha,m) + \delta \, p(\alpha,m) + \epsilon \, r(\alpha,m)}{\gamma \ + \ \delta \ + \ \epsilon}$$

$$e_r(\alpha,\rho) = \frac{\gamma \, o(\alpha,\rho) + \delta \, p(\alpha,\rho) + \epsilon \, r(\alpha,\rho)}{\gamma \ + \ \delta \ + \ \epsilon}$$

The factors γ, δ, and ϵ are used to define the importance of the obedience, pro-activeness, and results values respectively.

All these objective values provided by the reputation artifact can then be used by agents to compute the reputation of others. It is possible that in one organisation where violation is the rule, if you are a strong violator of norms, your reputation is perhaps greater that in an organisation where violation is not at all the rule.

6 Related works and discussion

Several proposals for organisational infrastructures have been proposed in the literature: MadKit, based on AGR organisational model [14]; Ameli [13] and Ameli$^+$ [16], based on Islander [12]; karma, based on TeamCore [30]; OperA [11]; $\mathcal{S}$-$\mathcal{M}$oise$^+$ [23], based on $\mathcal{M}$oise$^+$ [24]. In the sequel, these works and our proposal are discussed considering important topics and features.

Abstraction & encapsulation. The current OIs components are either in agents (Ameli, $\mathcal{S}$-$\mathcal{M}$oise$^+$, OperA) or services (MadKit). The approaches that use only services are not flexible enough to allow the management and change by the agents. Those that use only agents are using them for reactive and task oriented services. Some of those agents are not really pro-active and autonomous entities. In our framework, we raise the level of abstraction with respect to approaches in which organisation mechanisms are hidden at the implementation level. By using agents and artifacts, such mechanisms become parts of the agent world, suitably encapsulated in proper entities that agents then can inspect, reason and manipulate, by adopting a uniform approach.

Agent autonomy. All above mentioned OIs extinguish the agents' autonomy. In Ameli, for instance, the agents are autonomous to achieve goals but the communication is constrained (or regimented) by the OI; in $\mathcal{S}$-$\mathcal{M}$oise$^+$ the agents are autonomous concerning the communication protocols but constrained (or regimented) in the achievement and coordination of collective goals. In our proposal, agents are still autonomous with respect to decision of using or not a specific artifact – including the organisational artifacts – and keep their autonomy – in terms of control of their actions – while using organisational artifacts. Agents however can depend on the functionalities provided (encapsulated) by artifacts, which can concern, for instance, some kind of mediation with respect to the other agents co-using the same organisational artifact. Then, by enforcing some kind of mediation policy an artifact can be both an enabler and a constrainer of agent interactions. However, such a constraining function can take place without compromising the autonomy of the agents regarding their decisions. We also clearly consider two kinds of mechanisms to implement the norms: regimentations that are implemented in the artifacts and can not be violated and enforcement that are implemented both in the artifacts (the detection) and in the organisational agents (evaluation and judgement).

Distributed management. Some OI, as $\mathcal{S}$-$\mathcal{M}$oise$^+$ and MadKit, centralise all the management of the organisation in one agent or service bringing out scalability problems. Distributing the management of the organisation into different organisational artifacts realises a distributed coordination (meaning here more particularly synchronisation) of the different functions related to the management of the organisation. Completing this distribution of the coordination, the reasoning and decision processes which are encapsulated in the organisational agents may be also distributed among the different agents. Thanks to their respective autonomy, all the reasoning related to the management of the organisation (monitoring, reorganisation, control) may be decentralised into different loci of decision with a loosely coupled set of agents.

Openness. To be open to the entrance of heterogeneous agents is an important feature for MAS in general and a reason to establish an organisation for the system. This is thus also an issue considered by all above OIs. In most cases (e.g. $\mathcal{S}$-$\mathcal{M}$OISE^{+}, AMELI) , the agents have access to the organisational infrastructure by means of an agent communication language (KQML, FIPA-ACL) or other open protocols. ORA4MAS does not use a protocol or communication language; operations are used instead. The interaction between the agents and the organisation is no more expressed with an ACL semantic. Besides that, organisational artifacts, as any other kind of artifact, can be created and added dynamically as needed. They have a proper semantic description of both the functionalities and operating instructions, so conceptually agents can discover at runtime how to use them in the best way.
Still related to openness, the approach promotes heterogeneity of agent societies: artifacts can be used by heterogeneous kinds of agents, with different kinds of reasoning capabilities. Extending the idea to multiple organisations, we can have the same agents playing different roles in different organisations, and then interacting with organisational artifacts belonging to different organisations. The use of artifacts, and particularly the CARTAGO implementation, allows agents implemented in different languages to use the artifacts and cooperate using them. Most of the OI listed above give tools and support only for agents implemented in a particular language, normally Java — which is not the most appropriate language to code some types of agents.

'Organisational power back to agents'. The current implementations of OI conceive the organisation as a layer where the application agents relies on to participate in the organisation activities. The agents are not *actors* of this layer, they are simply passive users. This conception of OI is captured by the notion of regimentation and organisation artifacts in our proposal. However, our contribution in this context is to allow that some decisions that were embedded in the services go back to the agents' layer by means of organisational agents. In ORA4MAS artifacts encapsulate the coordination and synchronisation which were implemented in services. Control and judgement procedures are separated from these aspects and are embedded in organisational agents. Organisational agents can then use organisational artifacts to help them in deciding and eventually applying sanctions to other agents.

'Some answers to challenges raised in [1]' In [1] different challenges for building normative multiagent systems have been reported. We attempt in the fol-

lowing to position our work with respect to some of these challenges.
Challenge 1 and *Challenge 2* address respectively the need of tools for agents to support "communities in their task of recognizing, creating, and communicating norms to agents" and tools for agents "to simplify normative systems, recognize when norms have become redundant, and to remove norms". In the framework, the proposal of an OML used to declaratively represent an organisation at the three different levels are a step in the satisfaction of this need. Moreover, by providing OML embedding the expression of norms, these latter are anchored and contextualized within the organisation. Contrary to other approaches which hide the organisation entity, the artifacts building the organisation infrastructure of the framework propose a set of tools to act and manipulate this revealed organisational layer on which the organisation entity is deployed.
Few proposals of enforcement of norms are detailed in the context of an organisational infrastructure. This is also mentioned by the *Challenge 3* "Tools for agents to enforce norms". In the proposed framework, the structuration of the organisational artifacts and agents makes a clear distinction between *enforcement* and *regimentation*. Besides organisational agents, two special kinds of artifacts have been defined to address that challenge: NormativeBoard, ReputationBoard. In the same trend, the distinction complemented by the fact that the framework doesn't modify the agents internal decision proposes a clear basis to address the *Challenge 4* by developping "Tools for agents to preserve their autonomy". Work realised in [5] proposes a good starting point in that direction.

7 Conclusion

In this paper, we have proposed a framework for normative multiagent organisations. It is composed of an organisation modeling language in which norms can be expressed, organisation-awareness mechanisms that are under development and an organisation infrastructure which is based on the A&A meta-model. This latter is composed of a set of organisational artifacts that encapsulate the functional aspects of an organisation and organisation management and regulation. Organisational agents complement this overall picture by encapsulating the decision and reasoning side of the management of organisations and enforcement of norms.

Although we already have some initial results on the use of this framework, some extensions aim at taking advantage of the uniform concepts used to implement the environment and the organisation abstractions through the concept of artifacts. Such an homogeneous conceptual point of view will certainly help us to situate organisations in environment or to install the access to the environment into organisational models (in the same direction as proposed by [27]). Other points of investigation are (1) the study of the reorganisation process of a MAS using the ORA4MAS approach, (2) the impact of the reorganisation on the organisational artifacts, (3) the definition of a meta-organisation for the ORA4MAS, so that we have special roles for organisational agents that give them access to the organisational artifacts.

Acknowledgements

We would like to thank Alessandro Ricci, Michele Piunti and Rosine Kitio for their valuable work within ORA4MAS.

References

1. Guido Boella, Leendert Torre, and Harko Verhagen. Introduction to the special issue on normative multiagent systems. *Autonomous Agents and Multi-Agent Systems*, 17(1):1–10, 2008.
2. Olivier Boissier, Jomi Fred Hübner, and Jaime Simão Sichman. Organization oriented programming from closed to open organizations. In Gregory O'Hare, Michael O'Grady, Oguz Dikenelli, and Alessandro Ricci, editors, *Engineering Societies in the Agents World VII (ESAW 06)*, volume 4457 of *LNCS*, pages 86–105. Springer-Verlag, 2007.
3. Rafael H. Bordini, Jomi Fred Hübner, and Michael Wooldrige. *Programming Multi-Agent Systems in AgentSpeak using Jason.* Wiley Series in Agent Technology. John Wiley & Sons, 2007.
4. Jan Broersen, Mehdi Dastani, Joris Hulstijn, Zisheng Huang, and Leendert van der Torre. The BOID architecture: conflicts between beliefs, obligations, intentions and desires. In Jörg P. Müller, Elisabeth Andre, Sandip Sen, and Claude Frasson, editors, *Proceedings of the Fifth International Conference on Autonomous Agents*, pages 9–16, Montreal, Canada, 2001. ACM Press.
5. Cosmin Carabelea. *Reasoning about autonomy in open multi-agent systems - an approach based on the social power theory.* in french, ENS Mines Saint-Etienne, December 2007.
6. Cristiano Castelfranchi, Frank Dignum, Catholijn M. Jonker, and Jan Treur. Deliberate normative agents: Principles and architecture. In Nicholas R. Jennings and Yves Lespérance, editors, *Intelligent Agents VI, Agent Theories, Architectures, and Languages (ATAL), 6th International Workshop, ATAL '99, Orlando, Florida, USA, July 15-17, 1999, Proceedings*, volume 1757 of *LNCS*, pages 364–378. Springer, 2000.
7. Rosaria Conte and Mario Paolucci. *Reputation in Artificial Societies: Social Beliefs for Social Order.* Kluwer, 2002.
8. Luciano Coutinho, Jaime Sichman, and Olivier Boissier. Organizational modeling dimensions in multiagent systems. In V. Dignum, F. Dignum, and E. Matson, editors, *Proceedings of Workshop Agent Organizations: Models and Simulations (AOMS@IJCAI 07)*, 2007.
9. Mehdi Dastani. 2APL: a practical agent programming language. *Autonomous Agent and Multi-Agent Systems*, 16:241–248, 2008.
10. Virginia Dignum and Frank Dignum. Modeling agent societies: Co-ordination frameworks and institutions. In Pavel Brazdil and Alípio Jorge, editors, *Proceedings of the 10th Portuguese Conference on Artificial Intelligence (EPIA'01)*, LNAI 2258, pages 191–204, Berlin, 2001. Springer.
11. Virginia Dignum, Javier Vazquez-Salceda, and Frank Dignum. OMNI: Introducing social structure, norms and ontologies into agent organizations. In Rafael H. Bordini, Mehdi Dastani, Jürgen Dix, and Amal El Fallah-Seghrouchni, editors, *Proceedings of the Programming Multi-Agent Systems (ProMAS 2004)*, LNAI 3346, Berlin, 2004. Springer.

12. Marc Esteva, David de la Cruz, and Carles Sierra. ISLANDER: an electronic institutions editor. In Cristiano Castelfranchi and W. Lewis Johnson, editors, *Proceedings of the First International Joint Conference on Autonomous Agents and MultiAgent Systems (AAMAS 2002)*, LNAI 1191, pages 1045–1052. Springer, 2002.
13. Marc Esteva, Juan A. Rodríguez-Aguilar, Bruno Rosell, and Josep L. Arcos. AMELI: An agent-based middleware for electronic institutions. In Nicholas R. Jennings, Carles Sierra, Liz Sonenberg, and Milind Tambe, editors, *Proceedings of the Third International Joint Conference on Autonomous Agents and Multi-Agent Systems (AAMAS'2004)*, pages 236–243, New York, 2004. ACM.
14. Jacques Ferber and Olivier Gutknecht. A meta-model for the analysis and design of organizations in multi-agents systems. In Yves Demazeau, editor, *Proceedings of the 3rd International Conference on Multi-Agent Systems (ICMAS'98)*, pages 128–135. IEEE Press, 1998.
15. Nicoletta Fornara and Marco Colombetti. Specifying and enforcing norms in artificial institutions. In A. Omicini, B. Dunin-Keplicz, and J. Padget, editors, *Proceedings of the 4th European Workshop on Multi-Agent Systems (EUMAS 06)*, 2006.
16. Andrés García-Camino, J.A. Rodríguez-Aguilar, and Wamberto W. Vasconcelos. A distributed architecture for norm management in multi-agent systems. In Jaime Sichman, P. Noriega, J. Padget, and Sascha Ossowski, editors, *Coordination, Organizations, Institutions, and Norms in Agent Systems III*, volume 4870 of *LNAI*, pages 275–286. Springer, 2007. Revised Selected Papers.
17. Benjamin Gâteau, Olivier Boissier, Djamel Khadraoui, and Eric Dubois. Controlling an interactive game with a multi-agent based normative organisational model. In Noriega P., Vázquez-Salceda J., G. Boella, Boissier O., Dignum V., Fornara N., and Matson E., editors, *AAMAS 2006 and ECAI 2006 International Workshops, COIN 2006 Hakodate, Japan, May 9, 2006 Riva del Garda, Italy, August 28, 2006*, volume 4386 of *LNAI*, pages 86–100. Springer, 2007. Revised Selected Papers.
18. Lou Goble and John-Jules Ch. Meyer, editors. *Proceedings of the 8th International Workshop on Deontic Logic in Computer Science, DEON 2006, Utrecht, The Netherlands, July 12-14, 2006*, volume 4048 of *Lecture Notes in Computer Science*. Springer, 2006.
19. Davide Grossi, Huib Aldewered, and Frank Dignum. *Ubi Lex, Ibi Poena*: Designing norm enforcement in e-institutions. In P. Noriega, J. Vázquez-Salceda, G. Boella, O. Boissier, V. Dignum, N. Fornara, and E. Matson, editors, *Coordination, Organizations, Institutions, and Norms in Agent Systems II*, volume 4386 of *LNAI*, pages 101–114. Springer, 2007. Revised Selected Papers.
20. Jomi F. Hübner, Olivier Boissier, Rosine Kitio, and Alessandro Ricci. Instrumenting multi-agent organisations with organisational artifacts and agents: "giving the organisational power back to the agents". *Journal of Autonomous Agents and Multi-Agent*, 2009. To Appear.
21. Jomi Fred Hübner. *Um Modelo de Reorganização de Sistemas Multiagentes*. PhD thesis, Universidade de São Paulo, Escola Politécnica, 2003.
22. Jomi Fred Hübner, Olivier Boissier, and Laurent Vercouter. Instrumenting multi-agent organisations with reputation artifacts. In Jomi F. Hübner, Eric Matson, Olivier Boissier, and Virginia Dignum, editors, *Coordination, Organizations, Institutions, and Norms in Agent Systems IV*, volume 5428 of *LNAI*, pages 96–110. Springer, 2009.
23. Jomi Fred Hübner, Jaime Simão Sichman, and Olivier Boissier. S-MOISE+: A middleware for developing organised multi-agent systems. In Olivier Boissier, Virginia

Dignum, Eric Matson, and Jaime Simão Sichman, editors, *Coordination, Organizations, Institutions, and Norms in Multi-Agent Systems*, volume 3913 of *LNCS*, pages 64–78. Springer, 2006.

24. Jomi Fred Hübner, Jaime Simão Sichman, and Olivier Boissier. Developing organised multi-agent systems using the MOISE+ model: Programming issues at the system and agent levels. *International Journal of Agent-Oriented Software Engineering*, 1(3/4):370–395, 2007.
25. Rosine Kitio, Olivier Boissier, Jomi Fred Hübner, and Alessandro Ricci. Organisational artifacts and agents for open multi-agent organisations: "giving the power back to the agents". In Jaime Sichman, P. Noriega, J. Padget, and Sascha Ossowski, editors, *Coordination, Organizations, Institutions, and Norms in Agent Systems III*, volume 4870 of *LNCS*, pages 171–186. Springer, 2008. Revised Selected Papers.
26. M. Luck, P. McBurney, O. Shehory, and S. Willmott. *Agent Technology: Computing as Interaction (A Roadmap for Agent Based Computing)*. AgentLink, 2005. `http://www.agentlink.org/roadmap`.
27. F. Y. Okuyama, R. H. Bordini, and A. C da Rocha Costa. Spatially distributed normative objects. In P. Noriega, J. Vázquez-Salceda, G. Boella, O. Boissier, V. Dignum, N. Fornara, and E. Matson, editors, *Coordination, Organizations, Institutions, and Norms in Agent Systems II*, volume 4386 of *LNAI*, pages 133–146, 2007.
28. M. Piunti, A. Ricci, L. Braubach, and A. Pokahr. Goal-directed interactions in artifact-based mas: Jadex agents playing in cartago environments. In *IEEE/WIC/ACM Conferences on Web Intelligence and Intelligent Agent Technology (IAT-2008)*. IEEE/WIC/ACM, 2008.
29. Alexander Pokahr, Lars Braubach, and Winfried Lamersdorf. Jadex: A BDI reasoning engine. In Rafael H. Bordini, Mehdi Dastani, Jürgen Dix, and Amal El Fallah Seghrouchni, editors, *Multi-Agent Programming: Languages, Platforms, and Applications*, number 15 in Multiagent Systems, Artificial Societies, and Simulated Organizations, chapter 6, pages 149–174. Springer, 2005.
30. David V. Pynadath and Milind Tambe. An automated teamwork infrastructure for heterogeneous software agents and humans. *Autonomous Agents and Multi-Agent Systems*, 7(1-2):71–100, 2003.
31. Alessandro Ricci, Michele Piunti, L. Daghan Acay, Rafael H. Bordini, Jomi F. Hübner, and Mehdi Dastani. Integrating heterogeneous agent programming platforms within artifact-based environments. In Lin Padgham, David C. Parkes, Jörg Müller, and Simon Parsons, editors, *7th International Joint Conference on Autonomous Agents and Multiagent Systems (AAMAS 2008), Estoril, Portugal, May 12-16, 2008*, pages 225–232. IFAAMAS, 2008.
32. Alessandro Ricci, Mirko Viroli, and Andrea Omicini. CArtAgO: A framework for prototyping artifact-based environments in MAS. In Danny Weyns, H. Van Dyke Parunak, and Fabien Michel, editors, *Environments for MultiAgent Systems III*, volume 4389 of *LNAI*, pages 67–86. Springer, May 2007. 3rd International Workshop (E4MAS 2006), Hakodate, Japan, 8 May 2006. Selected Revised and Invited Papers.
33. Alessandro Ricci, Mirko Viroli, and Andrea Omicini. The A&A programming model & technology for developing agent environments in MAS. In Mehdi Dastani, Amal El Fallah-Seghrouchni, Alessandro Ricci, and Michael Winikoff, editors, *Programming Multi-Agent Systems, 5th International Workshop, ProMAS 2007,*

Honolulu, HI, USA, May 15, 2007, Revised and Invited Papers, volume 4908 of *LNCS*, pages 89–106. Springer, 2008.

34. Raimo Tuomela and Maj Bonnevier-Tuomela. Norms and agreement. *European Journal of Law, Philosophy and Computer Science 5*, pages 41–46, 1995.
35. J. Vázquez-Salceda, H. Aldewereld, and F. Dignum. Norms in multiagent systems: some implementation guidelines. In *Proceedings of the Second European Workshop on Multi-Agent Systems (EUMAS 2004)*, 2004. `http://people.cs.uu.nl/dignum/papers/eumas04.PDF`.

A modal logic for reasoning on consistency and completeness of regulations

C. Garion[1] and S. Roussel[1,2] and L. Cholvy[2]

[1] ISAE
10 avenue Edouard Belin,
31055 Toulouse, France
[2] ONERA Centre de Toulouse
2 avenue Edouard Belin
31055 Toulouse, France

Abstract. In this paper, we deal with regulations that may exist in multiagent systems in order to regulate agent behaviour. More precisely, we discuss two properties of regulations, consistency and completeness. After defining what consistency and completeness mean, we propose a way to consistently complete incomplete regulations. This contribution considers that regulations are expressed in a first order deontic logic.

1 Introduction

In a society of agents, a regulation is a set of statements, or norms, which rule the behaviour of agents by expressing what is obligatory, permitted, forbidden and under which conditions. Such a regulation is for instance the one which applies in most countries in EU: *smoking is forbidden in any public area except specific places and in such specific places, smoking is permitted.* Another example of regulation is the one which gives the permissions, prohibitions (and sometimes the obligations) of the different users of a computer system for file reading, file writing and file execution. Regulations are means to regulate agent behaviour so that they can live together. But in order to be useful, regulations must be *consistent* and, in most cases, they must also be *complete*.

Consistency is a property of regulations that has already been given some attention in the literature. For instance, as for confidentiality policies, consistency allows to avoid cases when the user has both the permission and the prohibition to know something [2]. More generally, according to [4] which studies consistency of general kind of regulations, a regulation is consistent if there is no possible situation which leads an agent to *normative contradictions* or *dilemmas* also called in [20] *contradictory conflicts* (a given behaviour is prescribed and not prescribed, or prohibited and not prohibited) and *contrary conflicts* (a given behaviour is prescribed and prohibited). Following this definition, consistency of security policies has then been be studied in [5].

Completeness of regulations has received much less attention. [2] proposes a definition of completeness between two confidentiality policies (for each piece of

Dagstuhl Seminar Proceedings 09121
Normative Multi-Agent Systems
http://drops.dagstuhl.de/opus/volltexte/2009/1904

information, the user must have either the permission to know it or the prohibition to know it), definition which has been adapted in [8] for multilevel security policies.

More recently, we have studied the notion of completeness for particular regulations which are policies ruling information exchanges in a multiagent system [6]. A definition of incompleteness for such policies has been given and a way to reason with incomplete policies has been defined. The approach taken in this work was rather promising and we have extended it for general regulations in [7]. The formal language used in those papers is classical first-order logic (FOL) following the ideas developed in [4]. In particular, deontic notions (obligation, permission, prohibition) are represented using predicate symbols. Because this leads to a rather complicated partition of the language between deontic predicate symbols and predicate symbols representing objects properties, this approach can be criticized. Moreover, deontic notions are classically represented in modal logic since [19, 14]. This is the reason why, in this present paper, we aim at using first order modal logic [12] to express regulations in a more elegant manner. Our objective is thus to reformulate the work described in [7] in a first-order modal framework.

This paper is organised as follows. Section 2 presents the logical formalism used to express regulations, the definitions of consistency and completeness of regulations. Section 3 focuses on the problem of reasoning with an incomplete regulation. Following the approach that has led to the default logic [17] for default reasoning, we present defaults that can be used in order to complete an incomplete regulation. In section 4, we present a particular example of regulation, information exchange policy. Finally, section 5 is devoted to a discussion and extensions of this work will be mentioned.

2 Regulations

The basic formalism used to model regulations is SDL (Standard Deontic Logic), a propositional modal logic [3]. We extend SDL to FOSDL (First-Order Standard Deontic Logic) in order to be able to express complex regulations implicating several agents. This is done in the way developed in [12].

2.1 Language

The alphabet of FOSDL is based on the following sets of non logical symbols: a set $\mathcal{P}$ of predicate symbols, a set $\mathcal{F}$ of function symbols and a modality symbol O representing obligation. The set of functions with arity 0 is called the *constants set* denoted $\mathcal{C}$. We define also the following logical symbols: a set $\mathcal{V}$ of variable symbols, $\neg$, $\vee$, $\forall$, (and). We call a *term* a variable or the application of a function symbol to a term.

We will use roman uppercase letters as predicate symbols, roman lowercase letters as function symbols and $\{x_1, \ldots, x_i, \ldots\}$ as variable symbols.

Definition 1. *The formulae of FOSDL are defined recursively as follows:*

- *if $t_1, \ldots, t_n$ are terms and P a predicate symbol with arity n, then $P(t_1, \ldots, t_n)$ is a formula of FOSDL.*
- *if φ is a formula of FOSDL, then $O\varphi$ is a formula of FOSDL.*
- *if ψ_1 and ψ_2 are formulae of FOSDL and x_1 a variable symbol, then $\neg\psi_1$, $\psi_1 \vee \psi_2$, $\forall x_1\ \psi_1$ are formulae of FOSDL.*

If ψ_1, ψ_2 and ψ_3 are FOSDL formulae and x_1 is a variable symbol, we also define the following abbreviations: $\psi_1 \wedge \psi_2 \equiv \neg(\neg\psi_1 \vee \neg\psi_2)$, $\psi_1 \otimes \psi_2 \otimes \psi_3 \equiv (\psi_1 \wedge \neg\psi_2 \wedge \neg\psi_3) \vee (\neg\psi_1 \wedge \psi_2 \wedge \neg\psi_3) \vee (\neg\psi_1 \wedge \neg\psi_2 \wedge \psi_3)$, $\psi_1 \rightarrow \psi_2 \equiv \neg\psi_1 \vee \psi_2$, $\psi_1 \leftrightarrow \psi_2 \equiv (\neg\psi_1 \vee \psi_2) \wedge (\psi_1 \vee \neg\psi_2)$, $\exists x_1\ \psi_1 \equiv \neg\forall x_1\ \neg\psi_1$.

The modalities for permission, noted P, and prohibition, noted F, are defined from O in the following way:

$$F\varphi \equiv O\neg\varphi$$
$$P\varphi \equiv \neg O\varphi \wedge \neg O\neg\varphi$$

It must be noticed that our definition of permission does not correspond to the usual definition of permission defined in SDL. According to SDL, something is permitted if its negation is not obligatory. However, it has been shown by lawyers [13] that the cases where permission is bilateral (permission to do and permission not to do) are the only valid ones. If not bilateral, permission to do entails obligation to do[1]. Our definition of bilateral permission corresponds to the notion of *optionality*[15] (something is optional iff neither it or its negation is obligatory).

A formula of FOSDL without modality is said to be *objective*. A term of FOSDL without variable symbols is said to be *ground*. The set of all ground terms in FOSDL is said to be the Herbrand universe HU. A formula of FOSDL without variable is said to be *ground*. A formula of FOSDL without the $\vee$, $\wedge$, $\otimes$, $\rightarrow$ nor $\leftrightarrow$ connectives is said to be a *literal*. Finally, we will call a *ground substitution* any function $\chi\ : \mathcal{V} \rightarrow HU$. If $\varphi(x)$ is a FOSDL formula with free variable x, $\varphi(\chi(x))$ is the formula φ in which occurrences of x have been replaced by $\chi(x)$.

2.2 Semantics

Semantics for propositional modal logics are classically defined using Kripke models. Models are defined by a *frame* $\langle \mathcal{W}, \mathcal{R} \rangle$, where $\mathcal{W}$ is a set of worlds and $\mathcal{R}$ an accessibility relation between worlds, and a relation $\Vdash$ between worlds and propositional letters. In the first-order case, we define models using an *augmented* frame and a first-order interpretation instead of $\Vdash$.

The semantics of first-order languages is based on a set of symbols (the *objects of discourse*), called *the domain*. The domain represents the objects on which the predicates will be evaluated by opposition to terms which are purely mathematical notions. In the case of first-order modal logic, we have to choose between

[1] For instance, when smoking is permitted, this implies that not smoking is also permitted. If not, that would mean that smoking would be obligatory.

a *constant domain* augmented frame and a *varying domain* augmented frame. In the first case, the domain is fixed for all the worlds in $\mathcal{W}$, in the second case, each world of $\mathcal{W}$ can have its own domain. We choose here a constant domain. As we study norms concerning only fixed elements, this choice is intuitively justified[2].

Definition 2. *Let $\mathcal{W}$ be a set of worlds, $\mathcal{R}_O$ a relation on $\mathcal{W}^2$ and $\mathcal{D}$ a non empty set of symbols representing the domain, then $\langle \mathcal{W}, \mathcal{R}_O, \mathcal{D} \rangle$ is called a frame.*

To define a model, we have to define an first-order interpretation which is done classically.

Definition 3. *An interpretation $\mathcal{I}$ in a frame $\langle \mathcal{W}, \mathcal{R}_O, \mathcal{D} \rangle$ is an application such that:*

- *for all n-ary function symbol f in $\mathcal{F}$ and all world $w \in \mathcal{W}$, $\mathcal{I}(f, w)$ is a function $\mathcal{D}^n \to \mathcal{D}$* independent of the world *w;*
- *for all n-ary predicate symbol P in $\mathcal{P}$ and all world $w \in \mathcal{W}$, $\mathcal{I}(P, w)$ is a relation on $\mathcal{D}^n$.*

Notice that we impose a particular condition on the interpretation of functions: the interpretation of a given function f is the same in every world w of $\mathcal{W}$ (this is possible because we use constant domain frames). This restriction allows us to escape from complicated technical details[3], for instance predicate abstraction. See [12] for more details.

Definition 4. *A model $\mathcal{M}$ is a structure $\langle \mathcal{W}, \mathcal{R}_O, \mathcal{D}, \mathcal{I} \rangle$ where $\langle \mathcal{W}, \mathcal{R}_O, \mathcal{D} \rangle$ is a frame and $\mathcal{I}$ an interpretation on $\langle \mathcal{W}, \mathcal{R}_O, \mathcal{D} \rangle$.*

Finally, we only use a class of frames that capture the correct behaviour of the modal operator O by constraining the accessibility relation $\mathcal{R}_O$.

Definition 5. *A FOSDL model is a model $\langle \mathcal{W}, \mathcal{R}_O, \mathcal{D}, \mathcal{I} \rangle$ such that $\mathcal{R}_O$ is serial.*

In order to define a satisfiability relation between models and formulae, we have to define the *valuation* notion which maps variables to elements of $\mathcal{D}$:

Definition 6. *Let $\mathcal{D}$ be a domain. A valuation on $\mathcal{D}$ is a complete function $\mathcal{V} \to \mathcal{D}$. A valuation σ' is a x-variant of a valuation σ if σ and σ' are identical except on x.*

Let t be a term and $\mathcal{V}(t)$ the set of variables in t, $\chi(t)$ is the term t in which each x_i in $\mathcal{V}(t)$ has been replaced by $\chi(x_i)$.

[2] Notice that varying domain can be useful. For instance in the study of doxastic first-order modal logic, an agent can learn the existence of a particular object, or a new object can appear.

[3] The main problem is to be able to characterize the meaning of a formula such as $OF(c)$ where c is a constant: does it mean that "it is obligatory that the object represented by c *in the current world* has F property" or "it is obligatory that *in each world*, the object represented by c has the F property".

The satisfiability relation $\models$ is defined as follows:

Definition 7. *Let $\mathcal{M} = \langle \mathcal{W}, \mathcal{R}_O, \mathcal{D}, \mathcal{I} \rangle$ a FOSDL model, w a world of $\mathcal{W}$ and σ a valuation on $\mathcal{D}$. Then:*

- *if P is a n-ary predicate symbol and $t_1, \ldots, t_n$ are terms, then $\mathcal{M}, w \models_\sigma P(t1, \ldots, t_n)$ iff $\langle \mathcal{I}(\sigma(t_1), w), \ldots, \mathcal{I}(\sigma(t_n), w) \rangle \in \mathcal{I}(P, w)$.*
- *if ψ is a FOSDL formula, then $\mathcal{M}, w \models_\sigma \neg\psi$ iff $\mathcal{M}, w \not\models_\sigma \psi$.*
- *if ψ_1 and ψ_2 are FOSDL formula, then $\mathcal{M}, w \models_\sigma \psi_1 \vee \psi_2$ iff $\mathcal{M}, w \models_\sigma \psi_1$ or $\mathcal{M}, w \models_\sigma \psi_2$.*
- *if $O\varphi$ is a FOSDL formula, $\mathcal{M}, w \models_\sigma O\varphi$ iff for every $v \in \mathcal{W}$ such that $w\mathcal{R}_O v$ holds, $\mathcal{M}, v \models_\sigma \varphi$.*
- *if ψ is a FOSDL formula, $\mathcal{M}, w \models_\sigma \forall x\ \psi$ iff for all valuations σ' x-variant of σ, $\mathcal{M}, w \models_{\sigma'} \psi$.*

Let ψ be a FOSDL formula. If for all valuations σ $\mathcal{M}, w \models_\sigma \psi$, we will note $\mathcal{M}, w \models \psi$. If $\mathcal{M}, w \models \psi$ for all w in $\mathcal{W}$, we will note $\mathcal{M} \models \psi$. Finally, if $\mathcal{M} \models \psi$ for all FOSDL models $\mathcal{M}$, then we will note $\models \psi$.

2.3 Axiomatics

We will now define an axiom system for FOSDL following the approach presented in[12]. In the following, $\varphi(x)$ denotes a formula in which the variable x may have free occurrences. We will say that a free variable y is *substitutable* for x in $\varphi(x)$ if no free occurrence of x in $\varphi(x)$ is in the scope of $\forall y$ in $\varphi(x)$.

Definition 8 (Axioms). *The formulae of the following forms are axioms:*
(Taut) *all classical FOL tautologies*
(KO) $O(\varphi \to \psi) \to (O\varphi \to O\psi)$
(DO) $O\varphi \to \neg O \neg \varphi$
(Bar1) $O(\forall x\ \varphi) \to \forall x\ O\varphi$
(Bar2) $\forall x\ O\varphi \to O(\forall x\ \varphi)$

Definition 9 (Inference Rules).
(MP) $\dfrac{\varphi \quad \varphi \to \psi}{\psi}$
(Gen) $\dfrac{\varphi}{\forall x\ \varphi}$
(NO) $\dfrac{\varphi}{O\varphi}$

Proposition 1 (Validity and soundness). *The previous system is valid and sound w.r.t. FOSDL semantics.*

The proof is given in [12].

We will define a *proof* of φ from the set of formulae Σ, noted $\Sigma \vdash \varphi$, as a sequence of formulae such that each one of them is an axiom, a formula of Σ, or produced by the application of an inference rule on previous formula.

In the following, $\bot$ will denote every formula that is a contradiction and $\top$ will denote every formula that is a tautology.

2.4 Regulation and integrity constraints modelling

In this section we define the notion of regulation and integrity constraints. First, we define the notion of rule, which is the basic component of a regulation. In this definition, rules have a general form, in particular they can be conditional.

Definition 10. *A rule is a formula of FOSDL of the form* $\forall \overrightarrow{x}\ l_1 \vee \ldots \vee l_n$ *with* $n \geq 1$ *such that:*

1. l_n *is of the form* $O\varphi$ *or* $\neg O\varphi$ *where* φ *is an objective literal*
2. $\forall i \in \{1, \ldots, n-1\}$, l_i *is an objective literal or the negation of an objective literal*
3. *if* x *is a variable in* l_n, *then* $\exists i \in \{1, \ldots, n-1\}$ *such that* l_i *is a negative literal and contains the variable* x
4. $\forall \overrightarrow{x}$ *denotes* $\forall x_1 \ldots \forall x_m$ *where* $\{x_1, \ldots, x_m\}$ *is the set of free variables appearing in* $l_1 \wedge \ldots \wedge l_{n-1}$.

In this definition, constraints (1) and (2) allow rules to be conditionals of the form "if such a condition is true then something is obligatory (resp. permitted or forbidden)". Constraint (3) restricts rules to range-restricted formulae[4]. Finally, rules are *sentences*, i.e. closed formulae, as expressed by constraint (4).

Notice also that we restrict in the definition of rules the formulae that can be defined as obligatory in the regulation: only objective literals can be obligatory or not obligatory.

We will write $\forall \overrightarrow{x}\ l_1 \vee \ldots \vee l_{n-1} \vee P\varphi$ as a shortcut for the two rules $\{\forall \overrightarrow{x}\ l_1 \vee \ldots \vee l_{n-1} \vee \neg O\varphi, \forall \overrightarrow{x}\ l_1 \vee \ldots \vee l_{n-1} \vee \neg O \neg\varphi\}$.

Definition 11. *A regulation is a set of rules.*

Let us consider an example which will help us to illustrate our purpose all along section 2 and 3.

Example 1 *We consider a regulation which rules the behaviour of a driver in front of a traffic light.*

The language needed is defined as follows:

- *green, orange, red, car, truck, bike,* A *and* T *are* 0*-arity functions, i.e. constants.*
- x, y, z, i *and* t *are variables.*
- $D(.)$ *is a predicate symbol that indicates that a term is a driver.*
- $TL(.)$ *is a predicate symbol that indicates that a term is a traffic light.*
- $C(.,.)$ *is predicate symbol that takes for parameters a traffic light and a color and indicates the traffic light color.*

[4] Range-restricted formulae are a decidable subset of domain-independent formulae which have been proved to be the only first order formulae having a meaning in modelling [9]. Notice in particular that by definition of FOSDL language, all variables appearing in l_n are free in l_n.

- $V(.,.)$ *is a predicate symbol that takes for parameters a driver and the type of vehicle he drives.*
- $IFO(.,.)$ *is a predicate symbol that takes for parameters a driver and a traffic light and indicates that the vehicle driven by the driver is in front of the traffic light.*
- $Stop(.,.)$ *is a predicate symbol that takes a driver agent and a traffic light for parameters and that indicates that this agent stops in front of the traffic light.*

Let's now take the three rules (r_0)*: "When a car-driver is in front of a traffic light that is red, he has to stop"* (r_1)*: "When a car-driver is in front of a traffic light that is orange, it is permitted for him to stop"* (r_2)*: "When a car-driver is in front of a traffic light that is green, he must not stop". These rules can be modelled by :*

$(r_0)\forall x \forall t\ D(x) \wedge TL(t) \wedge V(x, car) \wedge C(t, red) \wedge IFO(x,t) \rightarrow OStop(x,t)$
$(r_1)\forall x \forall t\ D(x) \wedge TL(t) \wedge V(x, car) \wedge C(t, orange) \wedge IFO(x,t) \rightarrow PStop(x,t)$
$(r_2)\forall x \forall t\ D(x) \wedge TL(t) \wedge V(x, car) \wedge C(t, green) \wedge IFO(x,t) \rightarrow FStop(x,t)$

2.5 Consistency of regulations

We now define a first notion for regulations, *consistency*. Intuitively, we will say that a regulation is consistent iff we cannot derive from the regulation using the system defined in 2.3 inconsistencies like $OStop(x,t) \wedge FStop(x,t)$. Consistency of a regulation is evaluated under *integrity constraints*, i.e. a set of closed objective formulae which can represent for instance physical constraints or domain constraints. In the following, we will note such an integrity constraints set IC.

First, we will define consistency of a regulation in a particular *state of the world*. Intuitively, states of the world are syntactic representations of classical first-order interpretations. They can also be assimilated to classical Herbrand models.

Definition 12 (state of the world). *A state of the world s is a complete and consistent set of objective ground literals.*

A state of the world is a syntactical representation of a Herbrand interpretation. Thus, for any n-ary predicate symbol P, any ground terms $t_1, \ldots, t_n$ and any state of the world s, either $P(t_1, \ldots, t_n) \in s$ or $\neg P(t_1, \ldots, t_n) \in s$. In the following, when describing a state of the world, we will omit the negative literals for readability.

Definition 13. *Let IC be a set of integrity constraints and s a state of the world. s is* consistent *with IC iff $s, IC \not\vdash \bot$.*

Definition 14. *Let ρ be a regulation, IC a set of integrity constraints and s a state of the world consistent with IC. ρ is consistent according to IC in s iff $\rho, IC, s \not\vdash \bot$.*

Example 2 *Let us resume example 1. Let us consider that IC contains two constraints: (1) a traffic light has a unique color and this color can be green, orange or red, and (2) a driver drives one and only one type of vehicle. Thus* $IC = \{\forall t\ TL(t) \rightarrow C(t, \mathit{green}) \otimes C(t, \mathit{orange}) \otimes C(t, \mathit{red}), \forall x \forall y \forall z\ D(x) \wedge V(x, y) \wedge V(x, z) \rightarrow y = z\}^5$.

Let s be the state of the world $\{D(A),\ TL(T), \mathit{IFO}(A, T),\ V(A, \mathit{car}),\ C(T, \mathit{red})\}$.

First, s is such that $s, IC \not\vdash \bot$. Let us consider a regulation ρ that contains the three rules (r_0), (r_1) and (r_2). In this case, $\rho, IC, s \not\vdash \bot$ (because the only deontic literal that can be deduced from ρ, IC and s is $O\mathit{Stop}(A, T)$). Thus, ρ is consistent according to IC in s.

Definition 15 (consistency of a regulation). *Let ρ be a regulation and IC a set of integrity constraints. ρ is consistent according to IC iff for all states of the world s such that $s, IC \not\vdash \bot$ then $\rho, IC, s \not\vdash \bot$.*

2.6 Completeness of regulations

Informally, a regulation is totally complete as soon as it prescribes the behaviour any agent should have in any situation. We can wonder if this definition really makes sense: can or must a regulation take into account all possible situations? Thus, we suggest to define a partial completeness restricted to two ground formulae φ and ψ: φ represents a particular situation in which we want to evaluate the regulation and ψ a predicate ruled by the regulation. Thus, we want a regulation be complete for φ and ψ iff in any situation where φ is true, it is obligatory (resp. permitted, forbidden) that ψ.

This leads to the following definition:

Definition 16. *Let IC be a set of integrity constraints, ρ be a regulation consistent according to IC and s a state of the world consistent with IC. Let $\varphi(\overrightarrow{x})$ and $\psi(\overrightarrow{x})$ two objective formulae, $\overrightarrow{x}$ representing free variables in φ and $\psi(\overrightarrow{x})$ meaning that the free variables in ψ are a subset of $\overrightarrow{x}$. ρ is $(\varphi(\overrightarrow{x}), \psi(\overrightarrow{x}))$-*complete *according to IC in s for $\vdash$ iff for all ground substitutions χ such that $s \vdash \varphi(\chi(\overrightarrow{x}))$:*

$$\begin{gathered}\rho, s \vdash O\psi(\chi(\overrightarrow{x}))\ \mathit{or}\\ \rho, s \vdash F\psi(\chi(\overrightarrow{x}))\ \mathit{or}\\ \rho, s \vdash P\psi(\chi(\overrightarrow{x}))\end{gathered}$$

Example 3 *Let us consider the state of the world* $s_0 = \{D(A),\ TL(T), \mathit{IFO}(A, T),\ V(A, \mathit{Car}),\ C(T, \mathit{red})\}$. *Consider ρ and IC defined in example 2. s_0 is consistent with IC and $\rho, s \vdash O(\mathit{Stop}(A, T))$. Let's take $\varphi_0(x, t) \equiv TL(t) \wedge D(x) \wedge \mathit{IFO}(x, t)$ and $\psi_0(x, t) \equiv \mathit{Stop}(x, t)$. $s_0, IC \vdash \mathit{IFO}(A, T)$ and $\rho, IC, s_0 \vdash O(\mathit{Stop}(A, T))$. Thus, ρ is $(\varphi_0(x, t), \psi_0(x, t))$-complete according to IC in s_0 for $\vdash$.*

Let us now consider the state of the world $s_1 = \{D(A),\ TL(T), \mathit{IFO}(A, T),\ V(A, \mathit{Truck}),\ C(T, \mathit{red})\}$. *$s_1$ is consistent with IC. $s_1, IC \vdash \mathit{IFO}(A, T)$ but $\rho, IC, s_1 \not\vdash$*

[5] The introduction of equality is done in the same way as in [12].

$O\psi_0(A,T)$, $\rho, IC, s_1 \not\vdash P\psi_0(A,T)$ and $\rho, IC, s_1 \not\vdash F\psi_0(A,T)$. Thus, ρ is $(\varphi_0(x,t), \psi_0(x,t))$-incomplete according to IC in s_1 for $\vdash$. In fact, no rule of the regulation can be applied as the vehicle is not a car but a truck.

The previous definition can be generalized as follows:

Definition 17 (completeness of a regulation). *Let IC a set of integrity constraints and ρ be a regulation. Let $\varphi(\overrightarrow{x})$ and $\psi(\overrightarrow{x})$ be two objective formula with the same meaning as in definition 16. ρ is $(\varphi(\overrightarrow{x}), \psi(\overrightarrow{x}))$-*complete *according to IC for $\vdash$ iff for every state of the world s consistent with IC, ρ is $(\varphi(\overrightarrow{x}), \psi(\overrightarrow{x}))$-complete according to IC in s for $\vdash$.*

Completeness is an important issue for a regulation. For a given situation, without any behaviour stipulated, any behaviour could be observed and thus consequences could be quite important. With an incomplete regulation, we could (1) detect the "holes" of the regulation and send them back to the regulation designers so that they can correct them or (2) detect the "holes" of the regulation and apply on those holes some completion rules to correct them. The first solution could be quite irksome to be applied (the number of holes could be quite important and thus correct them one by one quite long). Therefore, we put in place the second solution.

3 Reasoning with incomplete regulations

3.1 Defaults for completing regulation

Reasoning with incomplete information is a classical problem in logic and artificial intelligence: can we infer something about an information that is not present in a belief base? Several approaches have been defined, but we are here interested in one: default reasoning. The principle of default reasoning is quite simple: if an information is not contradictory with the informations that can be classically deduced from the belief base, then we can deduced another information from the belief base. A classical example is the following: let us suppose that we believe that "*every bird flies*", that "*penguins do not fly*" and "*penguins are birds*". Of course, the representation of this set of formulae in FOL is inconsistent (a bird which is also a penguin flies and do not fly at the same time). In fact, the first rule "*every bird flies*" is a default: "*if a is bird and it is not inconsistent with the belief base that a flies, then a flies*"[6]. If a is a penguin, then "a flies" cannot be deduced, and it cannot be deduced that a is a penguin, then we can deduce that a flies.

Default logic is a non-monotonous extension of first-order logic introduced by Reiter [17] in order to formalize default reasoning. We will here follow the presentation of Besnard given in [1].

[6] Notice that in this case, the information that is not contradictory with the belief base and the information newly deduced are the same.

A default d is a configuration $\dfrac{P : J_1, \ldots, J_n}{C}$ where $P, J_1, \ldots, J_n, C$ are first-order closed sentences. P is called the *prerequisite* of d, $J_1, \ldots, J_n$ the *justification* of d and C the *consequence* of d. A default theory $\varDelta = (D, F)$ is composed of a set of objective closed formulae F (facts) and a set of defaults.

A default theory (D, F) can be given in a *surface form* (D', F) on condition that

$D = \{\dfrac{P(\overrightarrow{a}) : J_1(\overrightarrow{a}), \ldots, J_n(\overrightarrow{a})}{C(\overrightarrow{a})} \ : \ \dfrac{P(\overrightarrow{x}) : J_1(\overrightarrow{x}), \ldots, J_n(\overrightarrow{x})}{C(\overrightarrow{x})} \in D'$ and $\overrightarrow{a}$ is a ground term$\}$

and every element of D' is of the form $\dfrac{P(\overrightarrow{x}) : J_1(\overrightarrow{x}), \ldots, J_n(\overrightarrow{x})}{C(\overrightarrow{x})}$ where $P(\overrightarrow{x}), J_1(\overrightarrow{x}), \ldots, J_n(\overrightarrow{x}), C(\overrightarrow{x})$ are first-order sentences with free variables occurring in $\overrightarrow{x}$.

Using defaults we obtain *extensions*, i.e. sets of formulae that are deduced monotonically and non-monotonically from F. Let $\varDelta = (D, F)$ be a default theory where defaults contains only closed formulae, then a extension of $\varDelta$ is a set of formulae E verifying the following conditions:

1. $F \subseteq E$
2. $Th(E) = E$ where $Th(E) = \{\varphi \ : \ E \vdash \varphi\}$
3. if $\dfrac{P : J_1, \ldots, J_n}{C}$ is a default of D, then if $P \in E$ and J_1 is consistent with E, $\ldots$, J_n is consistent with E, then $C \in E$

Default theories can have many extensions or no extensions at all. Reiter showed in [17] that if F is consistent and if (D, F) has an extension, then this extension is consistent. He showed also that any normal and closed default theory has at least one extension.

Here, we are not interested in the fact that a given objective formula ψ is believed but in the fact that a given regulation deduces that it is obligatory, forbidden or tolerated (those cases are the only ones due to the D axiom of O). Thus, if the regulation is incomplete for an objective formula ψ (i.e. it does not deduce neither $O\psi$ nor $F\psi$ nor $P\psi$), then it can only be completed by assuming that $O\psi$ can be deduced, or $P\psi$, or $F\psi$. This leads to the three sets of defaults which are described in the following.

In the following, let IC be a set of integrity constraints, ρ be a consistent regulation according to IC and s be a state of the world consistent with IC. Let $\varphi(\overrightarrow{x})$ and $\psi(\overrightarrow{x})$ be two objective formulae verifying definition 16.

Definition 18. *Let $E_F(\overrightarrow{x})$, $E_P(\overrightarrow{x})$ and $E_O(\overrightarrow{x})$ be three objective formulae such that their respective set of free variables is in $\overrightarrow{x}$. We define a set of configuration as follows:*

$(DF_{\varphi,\psi})$ $\dfrac{\varphi(\overrightarrow{x}) \wedge E_F(\overrightarrow{x}) \ : \ F\psi(\overrightarrow{x})}{F\psi(\overrightarrow{x})}$

$(DP_{\varphi,\psi})$ $\dfrac{\varphi(\overrightarrow{x}) \wedge E_P(\overrightarrow{x}) \ : \ P\psi(\overrightarrow{x})}{P\psi(\overrightarrow{x})}$

$(DO_{\varphi,\psi})$ $\dfrac{\varphi(\overrightarrow{x}) \wedge E_O(\overrightarrow{x}) \ : \ O\psi(\overrightarrow{x})}{O\psi(\overrightarrow{x})}$

A $(\varphi(\overrightarrow{x}), \psi(\overrightarrow{x}))$*-completeness default theory for* ρ *and* s *is a default theory* $\Delta_{\rho,s}(\varphi(\overrightarrow{x}), \psi(\overrightarrow{x}))$ *whose surface form is given by* $(\{DF_{\varphi,\psi}, DP_{\varphi,\psi}, DO_{\varphi,\psi}\}, \rho \cup s)$

We can complete an incomplete regulation so that $\psi(\overrightarrow{x})$ is forbidden ($DF_{\varphi,\psi}$), permitted ($DP_{\varphi,\psi}$) or obligatory ($DO_{\varphi,\psi}$) depending on $E_F(\overrightarrow{x})$, $E_P(\overrightarrow{x})$ and $E_O(\overrightarrow{x})$. Following Reiter, we define a new inference relation $\vdash_*$ defined as follows:

Definition 19. *Let* γ *be a formula of FOSDL.* $\rho, s \vdash_* \gamma$ *iff there is an extension* E_γ *of* $\Delta_{\rho,s}(\varphi(\overrightarrow{x}), \psi(\overrightarrow{x}))$ *such that* $\gamma \in E_\gamma$.

Moreover, we will note $Th_*(E) = \{\varphi \ : \ E \vdash_* \varphi$ and φ is closed$\}$.

Notice that we define here what Reiter calls an existential inference. There are of course other sorts of inference, for instance universal, but as we will show in section 3.2 we will obtain only one extension in the cases we are interested in, so the different kinds of inference are identical.

The next step is to define the conditions under which the regulation is complete and consistent with this new inference. This will be addressed in the next section.

3.2 Consistency and completeness of the completed regulation

First, we extend the definitions 15, 16 and 17 by using $\vdash_*$ instead of $\vdash$ in those definitions. To distinguish the new notions of consistency and completeness from the old ones, we will use $*$ as a prefix (for instance we will write "$*$-consistency") or write explicitly "for $\vdash_*$" (for instance, we will write "consistent for $\vdash_*$").

The main result about completeness and consistency of the regulation obtained by using the default theory defined previously is expressed by the following proposition.

Proposition 2. *Let us consider a set of integrity constraints IC, a regulation* ρ *consistent according to IC and a state of the world* s *consistent with IC and such that* $\rho \cup s$ *is consistent. Let* $\varphi(\overrightarrow{x})$ *and* $\psi(\overrightarrow{x})$ *be two objective formulae verifying definition 16 and* $\Delta_{\rho,s}(\varphi(\overrightarrow{x}), \psi(\overrightarrow{x}))$ *the corresponding default theory.*

The following propositions are equivalent:

1. *for every vector* $\overrightarrow{a}$ *of ground terms, if* $s \vdash \varphi(\overrightarrow{a})$, $\rho, s \not\vdash O\psi(\overrightarrow{a})$, $\rho, s \not\vdash P\psi(\overrightarrow{a})$ *and* $\rho, s \not\vdash F\psi(\overrightarrow{a})$ *(i.e.* ρ *is not* $(\varphi(\overrightarrow{a}), \psi(\overrightarrow{a}))$*-complete in s), then* $s \vdash E_F(\overrightarrow{a}) \otimes E_P(\overrightarrow{a}) \otimes E_F(\overrightarrow{a})$.
2. ρ *is consistent and* $(\varphi(\overrightarrow{x}), \psi(\overrightarrow{x}))$*-complete for* $\vdash_*$ *in* s.

This proposition characterizes necessary and sufficient conditions for the defaults to consistently complete an incomplete regulation. More precisely, this proposition says that if every time the regulation does not prescribe a behaviour one and only one E_i is true, then the defaults consistently complete the regulation (because one and only one default is applied for a particular $\psi(\overrightarrow{a})$).

Example 4 *Consider the state of the world* $s_1 = \{D(A), TL(T), IFO(A, T), V(A, truck), C(T, red)\}$ *from the last example.* ρ *is incomplete in* s_1 *for* $\varphi_0(x, t) \equiv D(A) \wedge TL(T) \wedge IFO(A, T)$ *and* $\psi_0(x, t) \equiv Stop(A, T)$ *in* s_1.

Let's take $E_F(x, t) = V(x, truck) \wedge C(t, green)$, $E_P(x, t) = V(x, truck) \wedge C(t, orange)$ *and* $E_O(x, t) = V(x, truck) \wedge C(t, red)$, *then* $s_1 \vdash E_O(A, T)$. *Thus,* ρ *is consistent and* $(\varphi_0(x, t), \psi_0(x, t))$*-complete for* $\vdash_*$ *in* s_1.

Even if this necessary and sufficient condition is interesting in theory, it is not really useful for practical purposes. In fact, to verify that this condition is satisfied, we would have to detect every "hole" in the regulation. This detection is an operation we want to avoid. Thus, we try to find more general conditions that are still sufficient but not necessary for the completion rules to consistently complete the regulation. We present two immediate corollaries of the previous definition.

Corollary 1. *If* $s \vdash \forall \overrightarrow{x}\ \varphi(\overrightarrow{x}) \rightarrow E_O(\overrightarrow{x}) \otimes E_F(\overrightarrow{x}) \otimes E_P(\overrightarrow{x})$ *then* ρ *is consistent and* $(\varphi(\overrightarrow{x}), \psi(\overrightarrow{x}))$*-complete according to IC for* $\vdash_*$ *in* s.

Example 5 *Consider the state of the world* $s_2 = \{D(A), TL(T), IFO(A, T), V(A, bike), C(T, red)\}$. s_2 *is consistent with* IC. *Consider the regulation defined in example 1.*

This time, let us consider $E_F(x, t) = C(t, green)$, $E_P(x, t) = C(t, orange)$ *and* $E_O(x, t) = C(t, red)$. $s_2 \vdash E_O(A, T)$. *Thus,* ρ *is* $*$*-consistent and* $*$*-complete for* $\varphi_0(x, t)$ *and* $\psi_0(x, t)$ *in* s_2. *But we also have* $s_1 \vdash E_O(A, T)$, *so* ρ *is* $*$*-consistent and* $(\varphi_0(x, t), \psi(x, t))$*-complete for* $\vdash_*$ *in* s_1. *Those more general* E_i *allow us to have a regulation complete for any type of vehicle.*

Corollary 2. *If* $IC \vdash \forall \overrightarrow{x}\ E_O(\overrightarrow{x}) \otimes E_F(\overrightarrow{x}) \otimes E_P(\overrightarrow{x})$ *then* ρ *is consistent and* $(\varphi(\overrightarrow{x}), \psi(\overrightarrow{x}))$*-complete according to IC for* $\vdash_*$.

Example 6 $IC \vdash \forall t\ C(t, red) \otimes C(t, green) \otimes C(t, orange)$. *Thus* ρ *is* $*$*-consistent and* $(\varphi_0(x, t), \psi_0(x, t))$*-complete for* $\vdash_*$.

IC specifies that a traffic light has one and only one color among three colors Red, Orange and Green. If there is one E_i *for each color, we are sure that whatever the situation is, we can apply one and only one default if there is a "hole" in the regulation.*

Another alternative would be to take fixed E_i. For example, we could take one E_i equal to $\top$ and the two others to $\bot$. We have three cases:

- suppose that $E_F \equiv \top$, $E_P \equiv \bot$ and $E_O \equiv \bot$. In this case, according to completion rules, everything that is not specified as obligatory or permitted by the regulation is forbidden. This strict behaviour could be observed for regulations that rule a highly secured system where each action has to be explicitly authorized before being performed.

- suppose that $E_F \equiv \bot$, $E_P \equiv \top$ and $E_O \equiv \bot$. We are here in the opposite situation, meaning that everything that is not obligatory or forbidden is permitted. This "tolerant" behaviour could be observed for regulations for dimmed secured systems where everything that is not forbidden or obligatory is implicitly permitted.
- suppose that $E_F \equiv \bot$, $E_P \equiv \bot$ and $E_O \equiv \top$. In this case, every action that is not forbidden or permitted has to be performed.

4 Examples of regulations: information exchange policies

An information exchange policy is a regulation which prescribes the behaviour of agents in a multiagent system regarding information communication. To describe such policies, we need five predicate symbols: *Agent*, *Info*, *Receive*, *Topic* and *Tell*. $Agent(x)$ means that x is an agent, $Info(i)$ means that i is an information, $Receive(x, i)$ means that agent x receives information i. $Topic(i, t)$ means that information i deals with topic t. $Tell(x, i, y)$ means that agent x tells agent y an information i. We also define constants a, b, i_1, *EqtCheck*, *ExpRisk*, *Meeting* and *EqtOutOfOrder*.

The consistency of such policies is defined by definition 14. The completeness of such policies is defined by instantiated definition 16 with the following specific formula:

$$\begin{aligned}\varphi(x, i, y) \equiv & Agent(x) \wedge Info(i) \wedge Receive(x, i) \wedge \\ & Agent(y) \wedge \neg(x = y) \\ \psi(x, i, y) \equiv & Tell(x, i, y)\end{aligned}$$

This leads to the following definition:

Definition 20. *Let IC a set of integrity constraints, s a state of the world consistent with IC and ρ a regulation consistent in s according to IC. ρ is complete according to IC in s for $\vdash$ iff for all ground substitution χ such that $s \vdash Agent(\chi(x)) \wedge Info(\chi(y)) \wedge Receive(\chi(x), \chi(i)) \wedge Agent(\chi(y)) \wedge \neg(\chi(x) = \chi(y))$:*

$$\begin{aligned}&\rho, s \vdash O\,Tell(\chi(x), \chi(i), \chi(y)) \text{ or} \\ &\rho, s \vdash F\,Tell(\chi(x), \chi(i), \chi(y)) \text{ or} \\ &\rho, s \vdash P\,Tell(\chi(x), \chi(i), \chi(y))\end{aligned}$$

Thus, default are the following:

$$(DF_{\varphi,\psi})\ \frac{\varphi(x, i, y) \wedge E_F(x, i, y) : F\,Tell(x, i, y)}{F\,Tell(x, i, y)}$$

$$(DP_{\varphi,\psi})\ \frac{\varphi(x, i, y) \wedge E_P(x, i, y) : P\,Tell(x, i, y)}{P\,Tell(x, i, y)}$$

$$(DO_{\varphi,\psi})\ \frac{\varphi(x, i, y) \wedge E_O(, i, y) : O\,Tell(x, i, y)}{O\,Tell(x, i, y)}$$

Results proved in section 3 remain valid. In particular, we still have the three cases:

- $E_F \equiv \top$, $E_P \equiv \bot$ and $E_O \equiv \bot$.
  This applies to highly secured multiagent systems in which any communication action should be explicitly obligatory or permitted before being performed.
- $E_F \equiv \bot$, $E_P \equiv \top$ and $E_O \equiv \bot$.
  This case applies to lowly secured systems in which any communication action which is not explicitly forbidden is permitted.
- $E_F \equiv \bot$, $E_P \equiv \bot$ and $E_O \equiv \top$.
  In this case, unless explicit mentioned, sending information is obligatory.

In order to illustrate this, consider the example of a firm in which there is a manager and two employees. Consider a policy π_0 with only one rule which states that "*Managers are required not to inform their employees about any equipment checking information*". The rule is modelled by[7]

$\forall x \forall i \forall y\ Manager(x) \wedge Employee(y) \wedge Receive(x, i) \wedge Topic(i, EqtChk) \rightarrow O\neg Tell(x, i, y)$

Let us consider $IC = \emptyset$ (there is no integrity constraints) and the state of the world $s_0 = \{Agent(a), Agent(b), Manager(a), Employee(b), Info(i_1), Topic(i_1, ExpRisk), Receive(a, i_1)\}$. In this situation, a is a manager and b an employee. a has received information i_1 whose topic is "Explosion Risk".

As π_0 contains only one rule and s_0 is consistent with IC, π_0 is consistent in s_0.

However we have $s_0 \vdash Agent(a) \wedge Info(i_1) \wedge Receive(a, i_1) \wedge Agent(b) \wedge \neg(a = b)$ but $\pi_0, s_0 \not\vdash O(Tell(a, i_1, b))$ and $\pi_0, s_0 \not\vdash P(Tell(a, i_1, b))$ and $\pi_0, s_0 \not\vdash F(Tell(a, i_1, b))$. Thus, π_0 is incomplete for $\vdash$.

Incompleteness comes from the fact that the policy prescribes the behaviour of the manager if he/she receives an information about "Equipment Verification" but it does not prescribe anything as for information about "Explosion Risk". The policy does not state what the manager should do when he/she receives information about "Risk Explosion".

In order to complete the previous policy, we could take:

$E_F(x, y, i) = Topic(i, EqtChk)$, $E_P(x, y, i) = \bot$ and $E_O(x, y, i) = Topic(i, ExpRisk)$. Such a choice forces the manager to tell its employees about "Risk Explosion" information. We can verify that π_0 is complete and consistent for $\vdash_*$ in s_0 for $\varphi(x, i, y)$ and $\psi(x, i, y)$.

Let consider now that IC contains the constraint "*An information has one and only one topic and this topic can be EqtChk, ExpRisk, Meeting or EqtOutOfOrder*". Take:

$$\begin{aligned} E_F(x, y, i) \equiv\ & Topic(i, EqtChk) \vee \\ & Topic(i, Meeting) \\ E_P(x, y, i) \equiv\ & Topic(i, EqtOutOfOrder) \\ E_O(x, y, i) \equiv\ & Topic(i, ExpRisk) \end{aligned}$$

We can apply the corollary 2 to conclude that π_0 is $*$-complete and $*$-consistent for $\varphi(x, i, y)$ and $\psi(x, i, y)$.

[7] The predicate names are obvious thus we do not formally define the language.

5 Conclusion

In this paper, we addressed the problem of analysing consistency and completeness of regulations which may exist in a society of agents in order to rule their behaviour.

More specifically, we have defined a modal logical framework and showed how to express a regulation within this framework. We then have reminded of a definition of consistency and we have defined what meant completeness for a regulation. The definition of completeness we gave is rather general. We also dealt with incomplete regulations and proposed a way for completing them by using defaults. We have established several results which show when these defaults consistently complete a regulation.

Although these notions (except defaults) were present in [6, 7], we have extended these previous papers in two points:

- first, we use a first-order modal logic to represent regulations. This allows us to clearly distinguish between the properties with which the deontic notions deal from the deontic notions and we keep the expressiveness of FOL for objects properties.
- second, the approach taken in the previous papers to complete a regulation was to extend the CWA (Closed World Assumption) defined by Reiter in order to complete first-order databases [16]. We choose here to use default reasoning, which is a more elegant solution to complete regulations.

The notion of completeness developed here is in fact a kind of local completeness, in the sense that we require to have $O(\psi(\overrightarrow{x}))$, $P(\psi(\overrightarrow{x}))$ or $F(\psi(\overrightarrow{x}))$ only for a specific context represented by formula $\varphi(\overrightarrow{x})$. That looks close to the notion of completeness introduced in the databases domain by [18, 10], who noticed that some of the integrity constraints that are expressed on a database are rules about what the database should know (i.e. these are rules about what should be deduced in the database). For instance, the integrity constraint expressing that "any employee has got a phone number, a fax number or a mail address" expresses in fact that, for any employee known by the database, the database knows its phone number, its fax number or its mail address[8]. As first mentioned by Reiter [18], this integrity constraint expresses a kind of local completeness of the database. Reiter's defaults can be used in order to complete such a database in case of incompleteness. For instance, one of the rules can be that if the database does contain any required information (no phone number, no fax number, no mail address) for a given employee but if the department that employee works in is known, then it can be assumed that its phone number is the phone number of its department.

Studying the formal link between the notion of completeness introduced in this paper and that notion of local completeness constitutes one interesting extension of this work.

[8] Notice that this does not prevent the fact that in the real world, an employee of the company has no telephone number, no fax number and no mail address

Furthermore, in order to deal with more general regulations, this present work must be extended. In particular, we have to extend it by considering more notions, among them time and action. Indeed, as it is shown in [11], the issue of time is very important when speaking about obligations and we will have to consider different types of time among which, at least, the time of validity of the norms and the deadlines beared on the obligations. Notice also for instance that in most of the examples of this paper, the predicates concerned by deontic operators represent actions (tell, stop etc.). The adding of a dynamic modal operator and/or temporal operator may be interesting. We will thus obtain a multimodal logic with strong expressiveness.

Finally, we developed a really simple model of the deontic notions by using SDL and lots of classical problem in deontic logic are not handled here: norms with exceptions, contrary-to-duties, collective obligations etc. Another extension of this work will be to define a logic that can deal with these problems.

References

1. P. Besnard. *An introduction to default logic.* Springer-Verlag, 1989.
2. P. Bieber and F. Cuppens. Expression of confidentiality policies with deontic logic. In *Deontic logic in computer science: normative system specification*, pages 103–121. John Wiley and Sons, 1993.
3. B. F. Chellas. *Modal logic, an introduction.* Cambridge University Press, 1980.
4. L. Cholvy. Checking regulation consistency by using SOL-resolution. In *International Conference on Artificial Intelligence and Law*, pages 73–79, 1999.
5. L. Cholvy and F. Cuppens. Analyzing consistency of security policies. In *1997 IEEE Symposium on Security and Privacy*, pages 103–112. IEEE, 1997.
6. L. Cholvy and S. Roussel. Reasoning with incomplete information exchange policies. In K. Mellouli, editor, *Symbolic and Quantitative Approaches to Reasoning with Uncertainty, 9th European Conference, ECSQARU'07*, number 4724 in Lecture Notes in Articial Intelligence, pages 683–694. Springer-Verlag, 2007.
7. L. Cholvy and S. Roussel. Consistency and completeness of regulations. In *Proceedings of he third International Workshop on Normative Multiagent Systems (NORMAS'08)*, pages 51–65, 2008.
8. F. Cuppens and R. Demolombe. A modal logical framework for security policies. In *Lectures Notes in Artificial Intelligence*, volume 1325, page 1997. Springer, 1997.
9. R. Demolombe. Syntactical characterization of a subset of domain independent formulas. *Journal of the Association for Computer Machinery*, 39(1):71–94, 1982.
10. R. Demolombe. Database validity and completeness: another approach and its formalisation in modal logic. In Enrico Franconi and Michael Kifer, editors, *Proc. of the 6th International Workshop on Knowledge Representation meets Databases (KRDB'99)*, pages 11–13. CEUR-WS.org, 1999.
11. R. Demolombe, P. Bretier, and V. Louis. Norms with deadlines in dynamic deontic logic. In Gerhard Brewka, Silvia Coradeschi, Anna Perini, and Paolo Traverso, editors, *Proceedings of ECAI 2006, 17th European Conference on Artificial Intelligence*, pages 751–752. IOS Press, 2006.
12. M. Fitting and R. L. Mendelsohn. *First-order modal logic.* Kluwer Academic, 1999.
13. C. Groulier. *Normes permissives et droit public.* PhD thesis, Université de Limoges, 2006. Available on `http://www.unilim.fr/scd/theses/accesdoc.html`. In French.

14. R. Hilpinen, editor. *Deontic logic*. Reidel Publishing Company, 1971.
15. P. Mc Namara. Deontic logic. Stanford Encyclopedia of Philosophy, `http://plato.stanford.edu/entries/logic-deontic/`.
16. R. Reiter. On closed world databases. In J. Minker J.-M. Nicolas H. Gallaire, editor, *Logic and Databases*. Plenum Publications, 1978.
17. R. Reiter. A logic for default reasoning. *Artificial Intelligence*, 13(1,2), 1980.
18. R. Reiter. What should a database know? *Journal of Logic Programming*, 14(1,2):127–153, 1992.
19. G. H. von Wright. Deontic logic. *Mind*, 60:1–15, 1951.
20. E. Vranes. The definition of "norm conflict" in international law and legal theory. *The European Journal of International Law*, 17(2):395–418, 2006.

A note on Brute vs. Institutional Facts: Modal Logic of Equivalence Up To a Signature

Davide Grossi

Institute of Logic, Language and Computation
University of Amsterdam
d.grossi@uva.lu

Abstract. The paper investigates the famous Searlean distinction between "brute" and "institutional" concepts from a logical point of view. We show how the partitioning of the non-logical alphabet—e.g., into "brute" and "institutional" atoms—gives rise to interesting modal properties. A modal logic, called UpTo-logic, is introduced and investigated which formalizes the notion of (propositional) logical equivalence *up to* a given signature.

1 Introduction

In the last decade the logical analysis of constitutive rules, initiated by [9], has focused on a number of aspects: defeasibility [3, 4], contextual and classificatory aspects [7, 8], mental aspects [12]. The prominent view has been to study constitutive rules, or "counts-as statements", as logical conditionals of the form $\varphi_1 \Rightarrow \varphi_2$ where the logic of $\Rightarrow$ was, from case to case, capturing the aforementioned aspects. One aspect, though, that has up to now been neglected concerns the different linguistic nature of the antecedent φ_1 and the consequent φ_2 of such conditionals.

According to Searle [14, 15] a characteristic aspect of constitutive rules is to link brute facts to institutional ones. Antecedent and consequent belong, somehow, to two different sets of concepts into which the language of institutions can be split. Institutional facts are constituted on the top of brute ones, giving to brute ones some sort of 'priority' upon the institutional ones.

The present paper explores, using modal logic, this linguistic aspect of constitutive rules. It develops ideas already introduced and partially investigated in [5,6]. It is structured as follows. Section 2 introduces the notion of equivalence *up to* a given propositional signature. Such notion is then semantically and axiomatically studied in a multi-modal language in Section 3. Section 4 discusses some related work and draws some conclusions.

2 Formal aspects of the brute vs. institutional distinction

In this section Searle's thesis concerning the distinction of brute and institutional facts is related to a specific notion of logical equivalence.

Dagstuhl Seminar Proceedings 09121
Normative Multi-Agent Systems
http://drops.dagstuhl.de/opus/volltexte/2009/1910

2.1 Counts-as conditionals, brute, and institutional facts

Let us start off with one of Searle's paradigmatic examples of a constitutive rule, the one concerning the institution of promising:

> Under certain conditions C anyone who utters the words (sentence) "I hereby promise to pay you, Smith, five dollars" promises to pay Smith five dollars [13, p. 44].

So, in context C the brute fact of uttering "I hereby promise" is a sufficient condition for the institutional fact of promising to occur. Following [7, 8] by interpreting contextual statements as forms of localized propositional validity, this can be semantically rendered as:

$$W_C \models \texttt{utter} \rightarrow \texttt{promise} \tag{1}$$

where W_C is the set of states modeling context C.[1] Now, `utter` belongs to the set BR of "brute" atoms, while `promise` to the set IN of "institutional ones". In the Spirit of Searle, sets BR and INS should obviously be taken to be disjoint, and to cover the set $\mathbf{P}$ of atoms of the language.

So where does the priority of BR in constituting the elements of IN arise? The thesis of this paper—already partially put forth in [6]—is that the priority of BR over IN consists in implications such as `utter` → `promise` in Formula 1 to cease to be valid once only the "brute" sublanguage, i.e., the atoms in BR, is considered. With respect to Formula 1, this means that counts-as conditionals imply the existence of a state w in context W_C and a state w' such that w and w' are indistinguishable from the point of view of BR (i.e., they satisfy the very same brute facts), and such that $W_C \cup \{w'\} \not\models \texttt{utter} \rightarrow \texttt{promise}$. If such a w' exists, then we can properly say that the truth of `promise` in W_C is constituted by the truth of `utter` since "all brute facts being equal" the implication possibly fails. The paper presents a logic to systematically handle this idea within a modal language.

2.2 Propositional equivalence *up to* a signature

The signature of a propositional language is its non-logial alphabet, that is, its set of propositional atoms. Let $\mathbf{P} = \{p, q, r \ldots\}$ be a countable set of propositional atoms, and let $\mathcal{L}(\mathbf{P})$ be the propositional language built on $\mathbf{P}$ and the usual Boolean connectives. We say that $\mathbf{P}$ is the signature of $\mathcal{L}(\mathbf{P})$.

Consider now the set $2^{\mathbf{P}}$ of all possible sub-signatures of $\mathcal{L}(\mathbf{P})$. Elements of such set will be denoted $P, Q, R, \ldots$ etc. Notice that the set of all sub-signatures of $\mathcal{L}(\mathbf{P})$ naturally yields a set algebra $\left\langle 2^{\mathbf{P}}, \cup, -, \mathbf{P}, \emptyset \right\rangle$. Two propositional models w and w' of $\mathcal{L}(\mathbf{P})$ are propositionally equivalent if they satisfy the same atoms in $\mathbf{P}$. As a consequence, for any formula φ of $\mathcal{L}(\mathbf{P})$: $w \models \varphi$ iff $w' \models \varphi$. If w and w' are equivalent ($w \sim w'$) then there is no set Φ of formulae of $\mathcal{L}(\mathbf{P})$ whose

[1] This is the semantics of what, in [7, 8], is called *classificatory counts-as*.

models contain w but not w', or vice versa. That is to say, the two models are indistinguishable for $\mathcal{L}(\mathbf{P})$.

However, two models which are not equivalent for $\mathbf{P}$ may be equivalent for some sub-signature $P \in 2^{\mathbf{P}}$. In this case, the two models cannot be distinguished by only looking at the atoms in P. The following definition makes such notion formal.

Definition 1. *(Equivalence up to a signature) Two models w and w' for a propositional language $\mathcal{L}$ are equivalent up to signature $P \in 2^{\mathbf{P}}$, or P-equivalent, if and only if for any $p \in P, w \models p$ iff $w' \models p$. If w and w' are P-equivalent we write $w \sim_P w'$.*

Obviously, if $w \sim_P w'$ then for all $\varphi \in \mathcal{L}(P)$: $w \models \varphi$ iff $w' \models \varphi$. The definition makes precise the idea of two propositional models agreeing up to what is expressible on a given signature.

Theorem 1. *(Properties of $\sim_P$) Let W be a set of models for the propositional language $\mathcal{L}(\mathbf{P})$. The following holds:*

(i) For every signature $P \in 2^{\mathbf{P}}$, the relation $\sim_P$ is an equivalence relation on W;
(ii) For all signatures $P, Q \in 2^{\mathbf{P}}$, if $P \subseteq Q$ then $\sim_Q \subseteq \sim_P$;
(iii) For each atom $p \in \mathbf{P}$, the relation $\sim_{\{p\}}$ yields a bipartition of W;
(iv) $\sim_{\mathbf{P}} = \sim$;
(v) $\sim_\emptyset = W^2$.

Proof. (i) The following holds: identity is a subrelation of $\sim_P$ for any sub-signature P; and that $\sim_P \circ \sim_P$ and $\sim_P^{-1}$ are subrelations of $\sim_P$ for any signature P. (ii) If $m \sim_Q m'$ then for all atoms $p \in Q$: $w \models p$ iff $w' \models p$. Therefore, since $P \subseteq Q$, $w \sim_P w'$. (iii) Suppose, per absurdum, that there exist three disjoint equivalence classes: $|w'|_{\sim_{\{p\}}}$, $|w''|_{\sim_{\{p\}}}$ and $|w'''|_{\sim_{\{p\}}}$. For bivalence, we have either $w' \models p$ or $w' \not\models p$. Suppose, without loss of generality, that $w' \models p$. By Definition 1 it follows that $w'' \not\models p$ and $w''' \not\models p$. Hence $|w''|_{\sim_{\{p\}}} = |w'''|_{\sim_{\{p\}}}$, which is impossible. (iv) The set $\mathbf{P}$ is the signature of the propositional language $\mathcal{L}(\mathbf{P})$, hence $\sim_{\mathbf{P}}$ is the propositional equivalence relation for $\mathcal{L}(\mathbf{P})$. (v) Suppose, per absurdum, there exists $w, w' \in W$ such that not $w \sim_\emptyset w'$. For Definition 1, there exists $p \in \emptyset$ such that $w \models p$ and $w' \not\models p$ (or viceversa), which is impossible.

Besides showing that signature-based equivalence is an equivalence relation (i), Theorem 1 shows also that: (ii) the bigger the signature, the more fine-grained is the equivalence relation; (iii) equivalences based on singleton signatured partition the set of states in two classes; (iv) if the propositional language under consideration is $\mathcal{L}(\mathbf{P})$ then relation $\sim_{\mathbf{P}}$ is standard propositional equivalence; (iv) $\sim_\emptyset$ is the universal relation on W. Notice also that from (ii) and (iii) follows that for every signature P it is the case that $\sim \subseteq \sim_P$, that is, propositional equivalence implies signature-based equivalence.

3 A modal logic of propositional equivalence *up to* a signature

The present section presents a modal logic—which we call UpTo—characterizing the notion of propositional equivalence *up to* a given signature.

3.1 Syntax of UpTo.

Let $\mathbf{P} = \{p, q, r \ldots\}$ be a countable set of propositional atoms. The language $\mathcal{L}_{\mathsf{UpTo}}(\mathbf{P})$[2] of logic UpTo on $\mathbf{P}$ is defined by the following BNF:

$$\mathcal{L}_{\mathsf{UpTo}} : \varphi ::= p \mid \neg\varphi \mid \varphi \wedge \varphi \mid [P]\varphi$$

where p ranges over $\mathbf{P}$ and P over $2^{\mathbf{P}}$. The Boolean connectives $\top, \vee, \rightarrow, \leftrightarrow$ and the dual operators $\langle P \rangle$ are defined as usual.

3.2 Semantics of UpTo.

Let us first define frames and models built on the notion of equivalente *up to* a given signature, in short, UpTo-frames and UpTo-models.

Definition 2. *(UpTo-frames) An UpTo-frame $\mathcal{F} = \langle W, \{\sim_P\}_{P \in 2^{\mathbf{P}}} \rangle$ for the propositional language $\mathcal{L}(\mathbf{P})$ is a tuple such that:*

- *W is a non-empty set of states;*
- *Each $\sim_P$ is an equivalence relation based on signature $P \in 2^{\mathbf{P}}$.*

Intuitively, an UpTo-frame fixes a particular arrangement of the equivalence classes available given a propositional language $\mathcal{L}(\mathbf{P})$. To make a simple example, suppose $W = \{w', w''\}$, $\mathbf{P} = \{p\}$ and $\sim_{\{p\}} = \{(w', w'), (w'', w'')\}$. Such frame for $\mathcal{L}(\{p\})$ states that w' and w'' are equivalent up to signature $\{p\}$ only to themselves. The valuation function will then say whether it is w' that satisfies p while w'' does not, or vice versa. This brings us to the notion of UpTo-model.

Definition 3. *(UpTo-models) An UpTo-model $\mathcal{M} = \langle \mathcal{F}, \mathcal{I} \rangle$ for the modal language $\mathcal{L}_{\mathsf{UpTo}}(\mathbf{P})$ is a tuple such that:*

- *$\mathcal{F}$ is an UpTo-frame for the propositional language $\mathcal{L}(\mathbf{P})$;*
- *$\mathcal{I} : \mathbf{P} \longrightarrow 2^W$ is an interpretation function.*

It may be instructive to notice that for each UpTo-frame there are exactly $2^{\mathbf{P}}$ different UpTo-models since Definition 1 requires that, for any atom p in $\mathbf{P}$, each element in the bipartition yielded by p coincides either with the truth-set of p or with the truth-set of $\neg p$.

The satisfaction relation is defined as follows.

Definition 4. *(Satisfaction for UpTo-models) Let $\mathcal{M}$ be an UpTo-model for $\mathcal{L}_{\mathsf{UpTo}}(\mathbf{P})$, $w \in W$ and $\varphi, \psi \in \mathcal{L}_{\mathsf{UpTo}}(\mathbf{P})$.*

$$\begin{aligned}
\mathcal{M}, w \models p &\quad \textit{iff} \quad w \in \mathcal{I}(p); \\
\mathcal{M}, w \models \neg\varphi &\quad \textit{iff} \quad \mathcal{M}, w \not\models \varphi; \\
\mathcal{M}, w \models \varphi \wedge \psi &\quad \textit{iff} \quad \mathcal{M}, w \models \varphi \;\&\; \mathcal{M}, w \models \psi; \\
\mathcal{M}, w \models [P]\varphi &\quad \textit{iff} \quad \forall w' \in W, w \sim_P w' : \mathcal{M}, w' \models \varphi
\end{aligned}$$

[2] In what follws we will often drop the reference to $\mathbf{P}$ and denote the language of UpTo simply by $\mathcal{L}_{\mathsf{UpTo}}$.

Formula φ is valid in $\mathcal{M}$, noted $\mathcal{M} \models \varphi$, if and only if for all w in W, $\mathcal{M}, w \models \varphi$. Formula φ is valid in $\mathcal{F}$, noted $\mathcal{F} \models \varphi$, if and only if it is valid in all models built on $\mathcal{F}$. Finally, φ is UpTo*-valid, noted $\models_{\mathsf{UpTo}} \varphi$, iff it is valid in all* UpTo*-frames. The logical consequence of formula φ from a set of formulae, noted $\Phi \models_{\mathsf{UpTo}} \varphi$, can be defined as usual.*

Intuitively, the *up to* operator $[P]$ means that φ holds in all states that are equivalent to the state of evaluation up to signature P.

3.3 Axiomatics of UpTo.

Logic UpTo is axiomatized by the following schemata.

$$
\begin{array}{rl}
(\mathtt{P}) & \text{all tautologies of propositional calculus} \\
(\mathtt{K}) & [P](\varphi \rightarrow \psi) \rightarrow ([P]\varphi \rightarrow [P]\psi) \\
(\mathtt{T}) & [P]\varphi \rightarrow \varphi \\
(4) & [P]\varphi \rightarrow [P][P]\varphi \\
(5) & \langle P\rangle\varphi \rightarrow [P]\langle P\rangle\varphi \\
(\mathtt{PO}) & [P]\varphi \rightarrow [Q]\varphi \text{ IF } P \subseteq Q \\
(\mathtt{Bipart}) & [\{p\}]p \vee [\{p\}]\neg p \\
(\mathtt{Dual}) & \langle P\rangle\varphi \leftrightarrow \neg[P]\neg\varphi \\
(\mathtt{MP}) & \text{IF } \vdash \varphi_1 \text{ AND } \vdash \varphi_1 \rightarrow \varphi_2 \text{ THEN } \vdash \varphi_2 \\
(\mathtt{N}) & \text{IF } \vdash \varphi \text{ THEN } \vdash [P]\varphi
\end{array}
$$

where P, Q range over $2^{\mathbf{P}}$, φ, ψ over $\mathcal{L}_{\mathsf{UpTo}}(\mathbf{P})$ and p over $\mathbf{P}$. The *up to* operators are **S5** operators with the addition of axioms PO (partial order) and Bipart (bipartition). Axiom PO orders the strength of the operators according to the relation of set-inclusion on the set of signatures. Notice that it consists of a transposition, in modal logic, of property (ii) in Theorem 1. Axiom Bipart states that if the signature considered consists of only atom p then it is either necessarily the case that p, or it is necessarily the case that $\neg p$. In other words, the equivalence up to p determines a bipartition of the set of states where the one cluster coincides with the set of p-states and the other with the set of $\neg p$ states. This axiom rephrases property (iii) of Theorem 1. Notice that from PO, Bipart and P follows that $[P]p \vee [P]\neg p$ if $p \in P$. [3]

Provability of a formula φ, noted $\vdash_{\mathsf{UpTo}} \varphi$, and derivability of a formula φ from a set of formulae Φ, noted $\Phi \vdash_{\mathsf{UpTo}} \varphi$ can be defined as usual. Appendix A offers a proof of the soundness and strong completeness of the proposed axiomatics with respect to the class of models built on UpTo-frames.

[3] A slightly different version of such schema has been used as an axiom in [5], where it is called NoCross. Notice that it forces the accessibility relation not to cross the bipartitions of the domain W yielded by each atom p, when p does belong to signature in the modal operator.

3.4 Embedding UpTo into S5

Take the standard modal language $\mathcal{L}_\Box(\mathbf{P})$ with one modal operator $\Box$ defined on the set of atoms $\mathbf{P}$. If we allow only *up to* operators $[P]$ where P is finite, it is possible to define an EXPtime truth-preserving reduction $f : \mathcal{L}_{\mathsf{UpTo}}(\mathbf{P}) \longrightarrow \mathcal{L}_\Box(\mathbf{P})$ as follows:

$$\begin{aligned}
f(p) &= p\\
f(\neg\varphi) &= \neg f(\varphi)\\
f(\varphi \wedge \psi) &= f(\varphi) \wedge f(\psi)\\
f([\emptyset]\varphi) &= \Box f(\varphi)\\
f([P]\varphi) &= \bigwedge_{\pi_i \in 2^P} \Big(\big(\bigwedge \pi_i^+ \wedge \bigwedge \pi_i^-\big) \to \Box\big(\big(\bigwedge \pi_i^+ \wedge \bigwedge \pi_i^-\big) \to f(\varphi)\big)\Big)
\end{aligned}$$

where $\pi_i^+ = \pi_i$ and $\pi_i^- = \{\neg p \mid p \in P \;\&\; p \notin \pi_i\}$. Intuitively, the *up to* operators are translated by taking care of all the possible truth-value combinations of the atoms in the signature P. If a given combination, e.g., $\bigwedge \pi_i^+ \wedge \bigwedge \pi_i^-$, is true at the given state, then in all accessible states, if that combination is true, than φ is also true. In addition, this should be the case for any combination drawn from a non-empty P, which explains $\bigwedge_{\pi_i \in 2^P - \emptyset}$. If P is empty, than $[P]$ is taken to be $\Box$. As a consequence, $\Box$ has to be interpreted as a universal modality (Theorem 1).

Theorem 2. *(f preserves satisfiability) Let $\mathcal{M} = \langle W, \{\sim_P\}_{P \in 2^{\mathbf{P}}}, \mathcal{I}\rangle$ be an* UpTo*-model for language $\mathcal{L}_{\mathsf{UpTo}}(\mathbf{P})$ and $\mathcal{M}' = \langle W', R', \mathcal{I}'\rangle$ be an* **S5** *model for $\mathcal{L}_\Box(\mathbf{P})$ such that:*

- $W' = W$;
- $R' = \sim_\emptyset$;
- $\mathcal{I}' = \mathcal{I}$.

For any $w \in W$ and $\varphi \in \mathcal{L}_{\mathsf{UpTo}}(\mathbf{P})$, $\mathcal{M}, w \models \varphi$ iff $\mathcal{M}', w \models f(\varphi)$.

Proof. The Boolean clauses and the clause for $[\emptyset]$ are obvious. As to the the last clause, by induction hypothesis (IH): $\mathcal{M}, w \models \varphi$ iff $\mathcal{M}', w \models f(\varphi)$. By IH, the semantics of $[P]$ and $\Box$, and Definition 1, the following expressions are all equivalent to $\mathcal{M}, w \models [P]\varphi$:

$$\forall w' \in W, w \sim_P w' : \mathcal{M}, w' \models \varphi$$

$$\forall w' \in W, w \sim_P w' : \mathcal{M}', w' \models f(\varphi)$$

$$\forall w' \in W, \forall \pi_i \in 2^P \text{ IF } \mathcal{M}', w \models \bigwedge \pi_i^+ \wedge \bigwedge \pi_i^- \text{ THEN } \mathcal{M}', w' \models \big(\bigwedge \pi_i^+ \wedge \bigwedge \pi_i^-\big) \to f(\varphi)$$

$$\forall \pi_i \in 2^P \text{ IF } \mathcal{M}', w \models \bigwedge \pi_i^+ \wedge \bigwedge \pi_i^- \text{ THEN } \mathcal{M}', w' \models \Box\Big(\big(\bigwedge \pi_i^+ \wedge \bigwedge \pi_i^-\big) \to f(\varphi)\Big)$$

$$\mathcal{M}', w' \models \bigwedge_{\pi_i \in 2^P - \emptyset} \Big(\big(\bigwedge \pi_i^+ \wedge \bigwedge \pi_i^-\big) \to \Box\big(\big(\bigwedge \pi_i^+ \wedge \bigwedge \pi_i^-\big) \to f(\varphi)\big)\Big)$$

This completes the proof.

As a consequence, we also obtain the following result.

Corollary 1. *(Decidability) The satisfiability problem for* UpTo *is decidable.*

Proof. The satisfiability problem for **S5** is decidable [2]. The result follows from Theorem 2.

Translation f makes explicit how the *up to* operators enable a compact representation of rather rich logical information. What can be expressed by UpTo can as well be expressed in **S5**, but not as easily.

4 Related work and conclusions

In these last two sections we relate the results presented in this paper to existing work in modal logic, and we finally draw some conclusions pointing at future research directions.

4.1 Related work: up to, release and ceteris paribus logics

The logic presented in Section 3 is a strict relative of the so-called release logics, first introduced and studied in [10, 11] in order to provide a modal logic characterization of a general notion of irrelevancy. Modal operators in release logics are **S5** operators indexed by an abstract set denoting the issues that are taken to be irrelevant while evaluating the formula in the scope of the operator. In [5] a special release logic is studied where the potentially irrelevant issues are precisely the propositional atoms of the language. This allows for the characterization of a notion of equivalence *modulo* a given signature. Instead of studying formulae $[P]\varphi$, whose intuitive meaning is "φ is the case" *up to* signature P, that logic studies formulae $[P]\varphi$ whose intuitive meaning is "φ is the case" *modulo* signature P, that is, if we abstract from the atoms in P. Therefore, in order to obtain a truth-preserving translation f of this logic to UpTo we just need to require: $f([P]\varphi) = [-P]f(\varphi)$, where $-$ is the set-theoretic complement. The UpTo logic can therefore be considered to belong to the family of release logics.[4]

Another work coming very close to the spirit of the present paper is [1]. In that paper a logic is presented for *ceteris paribus* preferences, that is to say, for preferences under the"all other things being equal" condition. Leaving the preferential component of such logic aside, its ceteris paribus fragment concerns sentences of the form $\langle \Gamma \rangle \varphi$ whose intuitive meaning is "there exists a state which is equivalent to the evaluation state with respect to all the formulae in the (finite) set Γ and which satisfies φ", where the formulae in Γ are drawn from the full language. At this point it is easy to see that logic UpTo is, in fact, the fragment of the ceteris paribus logic where Γ is allowed to consist only of a set of atoms. It is, we could say, the logic of "everything else being equal which you can express on this signature". From the semantic point of view, this means that UpTo-models contain considerably less equivalence classes than ceteris paribus models.

[4] See [5] for more details.

4.2 Conclusions

The paper has introduced and studied modal logic UpTo characterizing the notion of equivalence up to a given propositional signature. Soundness and completeness of the axiomatics, as well as the decidability of the satisfaction problem has been proven.

To conclude, let us go back to the beginning of Section 2 and show how Formula 1 can be appropriately extended in order to capture the "brute vs. institutional" distinction:

$$(2) \qquad W_C \models \mathtt{utter} \rightarrow \mathtt{promise} \text{ AND } \mathtt{W}_C \not\models [\mathtt{BR}](\mathtt{utter} \rightarrow \mathtt{promise})$$

Using the syntax of the modal context logic Cxt developed in [7, 8], Formula 2 could be expressed in the object-language as follows:

$$(3) \qquad [C](\mathtt{utter} \rightarrow \mathtt{promise}) \wedge \neg[\mathtt{C}][\mathtt{BR}](\mathtt{utter} \rightarrow \mathtt{promise})$$

where $[C]$ denotes the context operator. A systematic study of the interaction of logics Cxt and UpTo is left for future work.

References

1. J. van Benthem, P. Girard, and O. Roy. Everything else being equal: A modal logic for ceteris paribus preferences. *Journal of Philosophical Logic*, 38:83–125, 2009.
2. P. Blackburn, M. de Rijke, and Y. Venema. *Modal Logic*. Cambridge University Press, Cambridge, 2001.
3. G. Boella and L. Van der Torre. Regulative and constitutive norms in normative multiagent systems. In D. Dubois, C. A. Christopher A. Welty, and M. Williams, editors, *Proceedings of KR2004, Whistler, Canada*, pages 255–266, 2004.
4. G. Governatori, J. Gelati, A. Rotolo, and G. Sartor. Actions, institutions, powers. preliminary notes. In *International Workshop on Regulated Agent-Based Social Systems: Theories and Applications (RASTA'02)*, pages 131–147, 2002.
5. D. Grossi. Linguistic relevance in modal logic. In A. Nijholt, M. Pantic, M. Poel, and G. Hondorp, editors, *Proceedings of the 20th Belgian-Netherlands Conference on Artificial Intelligence (BNAIC'08)*. University of Twente, 2008.
6. D. Grossi. Pushing Anderson's envelope: The modal logic of ascription. In R. van der Meyden and L. van der Torre, editors, *Proceedings of the 9th International Conference on Deontic Logic in Computer Science (DEON 2008), Luxembourg, Luxembourg, July 15-18, 2008th International Workshop on Deontic Logic in Computer Science (DEON 2008), Luxembourg, Luxembourg, July 15-18, 2008*, number 5076/2008 in LNAI, pages 263–277. Springer, 2008.
7. D. Grossi, J.-J.Ch. Meyer, and F. Dignum. Classificatory aspects of counts-as: An analysis in modal logic. *Journal of Logic and Computation*, 16(5):613–643, 2006. Oxford University Press.
8. D. Grossi., J.-J.Ch. Meyer, and F. Dignum. The many faces of counts-as: A formal analysis of constitutive-rules. *Journal of Applied Logic*, 6(2):192–217, 2008.
9. A. J. I. Jones and M. Sergot. A formal characterization of institutionalised power. *Journal of the IGPL*, 3:427–443, 1996.

10. J. Krabbendam and J.-J. C. Meyer. Contextual deontic logics. In P. McNamara and H. Prakken, editors, *Norms, Logics and Information Systems*, pages 347–362, Amsterdam, 2003. IOS Press.
11. J. Krabbendam and J.-J.Ch. Meyer. Release logics for temporalizing dynamic logic, orthogonalising modal logics. In M. Barringer, M. Fisher, D. Gabbay, and G. Gough, editors, *Advances in Temporal Logic*, pages 21–45. Kluwer Academic Publisher, 2000.
12. E. Lorini and D. Longin. A logical account of institutions: from acceptances to norms via legislators. In G. Brewka and J. Lang, editors, *International Conference on Principles of Knowledge Representation and Reasoning (KR), Sidney, Australia, 16/09/08-19/09/08*, pages 38–48, http://www.aaai.org/Press/press.php, 2008. AAAI Press.
13. J. Searle. How to derive "ought" from "is". *The Philosophical Review*, 73(1):43–58, 1964.
14. J. Searle. *Speech Acts. An Essay in the Philosophy of Language*. Cambridge University Press, Cambridge, 1969.
15. J. Searle. *The Construction of Social Reality*. Free Press, 1995.

A Soundness and completeness of UpTo

Soundness is easily proven.

Theorem 3. *(Soundness of* UpTo*) For any $\varphi \in \mathcal{L}_{\mathsf{UpTo}}$, if $\vdash_{\mathsf{UpTo}} \varphi$ then $\models_{\mathsf{UpTo}} \varphi$.*

Proof. It is well-known that inference rules `MP` and `N` preserve validity on any class of frames, and that axioms `T`, `4` and `5` are valid on models built on equivalence relations[5]. The validity of `PO` and of `Bipart` follows from Theorem 1.

As to completeness, we make use of the standard canonical model technique.

Lemma 1. *Logic* UpTo *is strongly complete w.r.t. the class of* UpTo*-frames iff every* UpTo*-consistent set Φ of formulae is satisfiable on some model built on an* UpTo*-frame.*

Proof. From right to left we argue by contraposition. If UpTo is not strongly complete w.r.t. the class then there exists a set of formulae $\Phi \cup \{\varphi\}$ s.t. $\Phi \models_{\mathsf{UpTo}} \varphi$ and $\Phi \nvdash_{\mathsf{UpTo}} \varphi$. It follows that $\Phi \cup \{\neg\varphi\}$ is UpTo-consistent but not satisfiable on any UpTo-model. From left to right we argue per absurdum. Let us assume that $\Phi \cup \{\neg\varphi\}$ is UpTo-consistent but not satisfiable in any sublanguage equivalent model built on a frame in class UpTo. It follows that $\Phi \models_{\mathsf{UpTo}} \varphi$ and hence $\Phi \cup \{\neg\varphi\}$ is not UpTo-consistent, which is impossible.

Now let $\mathcal{M}^{\mathsf{UpTo}}$ be the canonical model of logic UpTo in language $\mathcal{L}_{\mathsf{UpTo}}(\mathbf{P})$. Model $\mathcal{M}^{\mathsf{UpTo}}$ is the structure $\left\langle W^{\mathsf{UpTo}}, \{R_P^{\mathsf{UpTo}}\}_{P \in \mathbf{P}}, \mathcal{I}^{\mathsf{UpTo}} \right\rangle$ where:

1. The set W^{UpTo} is the set of all maximal UpTo-consistent sets.
2. The canonical relations $\{R_P^{\mathsf{UpTo}}\}_{P \in \mathbf{P}}$ are defined as follows: for all $w, w' \in W^{\mathsf{UpTo}}$, if for all formulae φ, $\varphi \in w'$ implies $\langle P \rangle \varphi \in w$, then $w R_P^{\mathsf{UpTo}} w'$.
3. The canonical interpretation $\mathcal{I}^{\mathsf{UpTo}}$ is defined by $\mathcal{I}^{\mathsf{UpTo}}(p) = \{w \in W^{\mathsf{UpTo}} \mid p \in w\}$.

[5] See [2].

We have now to prove the Existence and Truth Lemmata for logic UpTo.

Lemma 2. *(Existence lemma) For all states in W^{UpTo}, if $\langle P\rangle\varphi \in w$ then there exists a state $w' \in W^{\mathsf{UpTo}}$ s.t. $R_P^{\mathsf{UpTo}}(w, w')$ and $\varphi \in w'$.*

Proof. The claim is proven by construction. Assume $\langle P\rangle\varphi \in w$ and let $w'_0 = \{\varphi\} \cup \{\psi \mid [P]\psi \in w\}$. The set w'_0 must be UpTo-consistent since otherwise there would exist $\psi_1, \dots, \psi_m \in w'_0$ such that $\vdash_{\mathsf{UpTo}} (\psi_1 \wedge \dots \wedge \psi_m) \to \neg\varphi$, from which we obtain $\vdash_{\mathsf{UpTo}} ([P]\psi_1 \wedge \dots \wedge [P]\psi_m) \to [P]\neg\varphi$. Since $[P]\psi_1, \dots, [P]\psi_m \in w$ we have that $\neg\langle P\rangle\varphi \in w$, which contradicts our assumption. Therefore, w'_0 is UpTo-consistent and can be extended to a maximal UpTo-consistent set (for Lindenbaum's Lemma[6]). By construction, w' contains φ and is such that for all ψ, if $[P]\psi \in w$ then w' contains ψ. From this it follows $R_P^{\mathsf{UpTo}}(w, w')$ since, if this was not the case, then there would exist a formula ψ' s.t. $\psi' \in w'$ and $\langle P\rangle\psi' \notin w$. Since w is maximal UpTo-consistent, $[P]\neg\psi' \in w$ and hence $\neg\psi' \in w'$, which contradicts the UpTo-consistency of w'.

Lemma 3. *(Truth lemma) For any formula $\varphi \in \mathcal{L}_{\mathsf{UpTo}}(\mathbf{P})$ and $w \in W^{\mathsf{UpTo}}$: $\mathcal{M}^{\mathsf{UpTo}}, w \models \varphi$ iff $\varphi \in w$.*

Proof. The claim is proven by induction on the complexity of φ. The Boolean case follows by the properties of maximal UpTo-consistent sets. As to the modal case, it follows from the definition of the canonical relations R_P^{UpTo} and Lemma 2.

Everything is now put into place to prove the strong completeness of UpTo.

Theorem 4. *(Strong completeness of UpTo) For any formula $\varphi \in \mathcal{L}_{\mathsf{UpTo}}(\mathbf{P})$ and set of formulae Φ, if $\Phi \vdash_{\mathsf{UpTo}} \varphi$ then $\Phi \models_{\mathsf{UpTo}} \varphi$.*

Proof. By Proposition 1, given an UpTo-consistent set Φ of formulae, it suffices to find a model state pair $(\mathcal{M}, w)$ such that: (a) $\mathcal{M}, w \models \Phi$, (b) $\mathcal{M}$ is an UpTo-model. Let $\mathcal{M}^{\mathsf{UpTo}} = \left\langle W^{\mathsf{UpTo}}, \{R_P^{\mathsf{UpTo}}\}_{P \in 2^{\mathbf{P}}}, \mathcal{I}^{\mathsf{UpTo}} \right\rangle$ be the canonical model of UpTo, and let Φ^+ be any maximal UpTo-consistent set in W^{UpTo} extending Φ. By Lemma 3 it follows that $\mathcal{M}^{\mathsf{UpTo}}, \Phi^+ \models \Phi$, which proves (a). To prove (b), we show that $\mathcal{M}^{\mathsf{UpTo}}$ is s.t.: (b.1) the frame on which $\mathcal{M}$ is based is an UpTo-frame; and (b.2) for all $p \in \mathbf{P}$, $R_{\{p\}}^{\mathsf{UpTo}}(w, w')$ iff it is the case that $p \in w$ iff $p \in w'$. As to (b.1), it is well-known that axioms `T`, `4` and `5` force the relations R_P^{UpTo} to be equivalence relations. It remains to be shown that if $P \subseteq Q$ then $R_Q^{\mathsf{UpTo}} \subseteq R_P^{\mathsf{UpTo}}$. Assume $R_Q^{\mathsf{UpTo}}(w, w')$. It follows that for all φ, if $\varphi \in w'$ then $\langle Q\rangle\varphi \in w$ and hence, by the contrapositive of axiom `P0`, $\langle P\rangle\varphi \in w$. Therefore, $R_P^{\mathsf{UpTo}}(w, w')$. As to (b.2), form left to right. Assume $R_{\{p\}}^{\mathsf{UpTo}}(w, w')$. For axioms `T` and `Bipart`, $p \in w$ iff $p \in w'$. From right to left, we assume $p \in w$ iff $p \in w'$. If $p \in w'$, by axioms `T` and `Bipart`, $\langle\{p\}\rangle p \in w$ and therefore $R_{\{p\}}^{\mathsf{UpTo}}(w, w')$. This completes the proof.

[6] See [2].

A Taxonomy for Ensuring Institutional Compliance in Utility Computing

Tina Balke

University of Bayreuth,
Chair for Information Systems Management,
Universitätsstr. 30, 95447 Bayreuth, Germany
http://www.bwl7.uni-bayreuth.de/en/index.html

Abstract. With the ongoing evolution from closed to open distributed systems and the lifting of the assumption that agents acting in such a system do not pursue own goals and act in the best interest of the society, new problems arise. One of them is that compliance cannot be assumed necessarily and consequently trust issues arise. One way of tackling this problem is by regulating the behavior of the agents with the help of institutions. However for institutions to function effectively their compliance needs to be ensured. Using a utility computing scenario as sample application, this paper presents a general applicable taxonomy for ensuring compliance that can be consulted for analyzing, comparing and developing enforcement strategies and hopefully will stimulate research in this area.

Key words: Institutions, Compliance, Enforcement, Regimentation, Norms, Sanctions, Utility Computing

1 Self-Interested Agents in Utility Computing

1.1 The Vision of Utility Computing

The vision of Utility Computing (UtiC) has gained significant interest in the last years and has become a popular buzzword. The word "utility" is used to make an analogy to the provision of other services, such as electrical power, the telephone, gas or water, in which the service providers seek to meet fluctuating customer needs, and charge for the fungible resources they sell based on usage rather than on a flat-rate basis[1]. In the computing context examples of such resources are storage space, server capacity, bandwidth or computer processing time. UtiC envisions that in contrast to traditional models of web hosting where

[1] It is important to note that although the services offered by the service providers are individualized, their basic components are very standardized resources that can be easily exchanged. Thus, a telephone provider for example may provide his customers with very different telephone packages, however the underlying resources he uses are standardized telephone units.

the web site owner purchases or leases a single server or space on a shared server and is charged a fixed fee, the fixed costs are substituted by variable costs and he is charged upon how many of the fungible resources he actually uses on demand over a given period of time in order to perform his computationally intensive calculations. The business idea behind this vision is that if a company has to pay only for what it is using it can adapt its cost structure and will be able to economize, i.e. save money, while the company offering utility computing resources can benefit from economies of scale by using the same infrastructure to serve multiple clients [8].

Looking at the nature of the resources sold in the UtiC context as well as the potential number of transactions that might be conducted in such an infrastructure, it seems reasonable to argue that UtiC is an ideal field of application for automated negotiations using artificial agents [16]. Thus, the resources traded in UtiC have a high degree of standardization, and furthermore the open interaction system as well as the high number of repetitive transaction, suggest the usage of artificial agents that act on behalf of their human owners. Furthermore, as mentioned in the AgentLink Roadmap [25], Multi-Agent-Systems (MAS) offer strong models for representing complex and dynamic environment that cannot be analyzed mathematically any more, but need to be simulated. However when thinking in the lines of this vision, several problems occur, such as the question about the risks involved in UtiC transactions. Thus, it has to be ascertained that the bilateral economic exchange envisioned in UtiC is very likely to involve risks, such as risks resulting from strategic- and parametric uncertainties, that shall be explained in the next section with regard to the problem of self-interested agents [38].

For the further analysis it has to be noted that this paper views UtiC as one possible field of application of electronic institutions or e-commerce. Nevertheless as it is a good example of an open distributed market that can be simulated with MAS simulation, it is explained in more detail at this point and will be used as example in the course of the paper.

1.2 Strategic Uncertainties resulting from Self-Interested Agents

As noted at the end of the last section, two main kinds of uncertainties exist in UtiC transactions, namely strategic- and parametric uncertainties. Whereas the latter ones refer to environmental uncertainties that cannot (or only with a disproportionate effort) be reduced by the UtiC participants, the strategic uncertainties concern the question of whether the transaction partners are willing to comply with what has been agreed on or not; and whether, if a transaction has had an adverse outcome, this was due to bad luck or bad intentions [26, 21]. Thus, if a buyer does not receive the promised UtiC resources from the seller, it is often hard to judge whether the seller did not deliver intentionally, or whether the transaction failed, because the network broke down for example.

The basic assumption behind this the problem of strategic uncertainty thereby is that agents are rational believe forming utility maximizing entities. Thereby it is assumed that the agents do not necessarily always act in the best interest of the societies global (or social) welfare. Instead they are likely to pursue their individual goals and try to maximize their profit (in terms of a maximization of their utility function) [31]. That is why agents may choose to not fulfill a contract as promised, if they expect a higher own utility from this. Thereby the decisions about the utility of different options by each agent are based on the limited information the particular agent has about the environment (i.e. the agents have a bounded rationality). As a result it becomes difficult to assess and control the utility functions of all participating agents. As a consequence it is very challenging from a UtiC environment designers point of view to control that the overall UtiC market outcome is as desired.

As a result institutions are needed that influence the utility functions of the agents and create incentives in such a way that cooperation is the dominant strategy and strategic uncertainty can be reduced to a minimum extent. In the next section the term institutions as used in this paper will be explained and the roles of institutions for regulating and controlling UtiC will be analyzed in more detail (2). Thereby special focus will be on the ensuring of the compliance with the institutions in UtiC as "if not being enforced effectively, [institutions] are nothing more than a decorative accessory" [9]. In the course of the analysis of the compliance-ensuring of institutions for UtiC, as the main contribution of this paper, in section 3 a taxonomy will be developed that tries to combine all elementary compliance-ensuring options in one table and to classify them in a expedient way. In a second step, the different elements of the taxonomy will be explained in detail in the sections 3.1–3.5 with the help of UtiC examples. Although, the main focus of this paper is UtiC, UtiC itself is just seen as a sample application for open distributed systems by the author. Thus, the author aims at presenting a general applicable taxonomy that can be consulted for analyzing, comparing and developing compliance-ensuring strategies and hopefully will stimulate research in this area. In a last step in this paper a research proposal will be made how to evaluate the taxonomy elements one against the other (chapter 4). Thereby 5 performance indicators will be presented that shall serve as a starting point for this analysis. Furthermore a research outline will of how the mechanisms shall be evaluated will be presented (chapter 4).

2 Ensuring Compliance of Institutions for Controlling Utility Computing

2.1 Institutions in Utility Computing

As mentioned in the last section, resulting from the openness of UtiC environments two problems arise: First of all anybody can participate in such an infrastructure and act intentionally and optimally towards their own specific goals (i.e utility functions). The second problem is that the overall social welfare of the system emerges as a result of the individual decisions and actions of the individual agents. However the utility functions that the agents base their decisions on are dynamic and normally private information of the individuals and are therefore hardly predictable for the UtiC environment designers. As a result, "appropriate" mechanisms that foster compliance and regulate the UtiC environments (in terms of defining a regulative framework as well as sanctions for non-compliance with the framework[2]) need to be applied in order to achieve an "acceptable" overall behavior. The most promising mechanism, which will be addressed in this paper is the usage of institutions. Institutions alter the relative prices for defections and thereby create incentives for a system-conform behavior. This paper thereby understands the term institutions as often used in new institutional economics, namely is follows:

> **Institutions** are formal (e.g. statute law, common law, regulations) and informal structures (e.g. conventions, norms of behavior and self imposed codes of conduct) and mechanisms of social order and cooperation governing the behavior of a set of individuals by attributing rights and obligations to them. They are identified with a social purpose and permanence, transcending individual intentions, and with the making and compliance-ensuring of rules governing cooperative human behavior and thereby define the social outcomes that result from individual actions [29, 33].

Looking at this definition three main aspects can be remarked. The first one is that in the institutional economic view institutions are understood as a very abstract notion of a set of norms or social structure. Hence norms are seen to be component of institutions, which are the overall concept of a regulative framework. The second aspect to be remarked concerns the role of institutions, namely the setting up of a framework of rules and actions in which the agents have to operate. This framework not only defines what agents should and should not do, but erects sanctions to be applied if the framework is violated. And this is where the third main aspect comes into play: the compliance-ensuring component. As North phrased it in [28] with regard to ensuring compliance:

> "...[it] poses no problem when it is in the interests of the other party to live up to agreements. But without institutional constraints, self-interested behavior will foreclose complex exchange, because of the uncertainty that the other party will find it in his or her interest to live up to the agreement."

What North formulated in this statement is very straight forward: the compliance with an institutional framework poses no problem, if no self-interested behavior is involved. If however – as in UtiC – this is not the case and agents can exhibit self-interested behavior, it is important that institutions do not only state a set of rules, but it needs to be taken care that their compliance is ensured, because otherwise the strategic uncertainties arising might negatively influence the usage of an environment (e.g. UtiC).

[2] In this paper special focus will be on regulative rules as they pose problems in terms of compliance-ensuring. Although being of high importance as well constitutive rules will be omitted as their non-compliance leads to nullity [20].

After this brief description of the role of institutions and especially the ensuring of their compliance in UtiC, in the next section, the related work relevant for the implementation of institutions in open distributed environments such as UtiC shall be reviewed. Thereby special focus will be once again on the compliance-ensuring aspect as it is has a key function in the success of every institutional setting.

2.2 Related Work

Already in 1998, Conte et al. [10] pointed out two distinct sets of problems relevant for MAS research on norms: (1) the interaction of different autonomous agents on a norm-governed basis and (2) the interaction of individual autonomous agents with the norms (including the acquisition and the violation of norms). This problem definition has been expanded by Boella and van der Torre [4] to include a third question that deals with the evolution of norms. The first of the three questions has been discussed at length by researchers using game-theoretic approaches [4]; however a model integrating these approaches with the different social, cognitive and normative concepts is still missing. The second question has been studied by Broersen et al. [7] for example, who focused on the agent architecture for determining how agents can acquire and violate norms and how norms in turn influence agent behavior. Last but not least, the third question has been dealt with by Verhagen [37] and some economic commerce researchers. Verhagen distinguished between norms created by legislators, norms negotiated between agents and norms emerging spontaneously and thereby laid the groundwork for a number of papers about protocols and social mechanisms for the creation [5] and agent mediated evolution of norms [34] in MAS. In spite of this intensive research on the creation of norms in MAS, little work has been done explicitly addressing the ensuring of the compliance with such norms. Thus, although trust and reputation mechanisms as centralized (e.g. eBay) and decentralized coordination and compliance-ensuring instances [32] have been discussed by a number of researchers, the mechanisms tend to concentrate on specific use cases and often fail to address the importance of these mechanisms in the compliance context. Thus, in many papers it is explicitly assumed, that all normative regulations can be asserted and therefore little or no thought is given on what happens if this assumption cannot be fulfilled, although many scientists have stated that institutions and norms are more or less senseless if their compliance cannot be ensured [9].

One of the few papers that deals with compliance and analyzes at what levels it can be applied in a system was written by Vázquez-Salceda et al. [36], who not only make a distinction between regimentation and enforcement, but also elaborate on the levels of observability of norm violations. Other authors that address the compliance topic in their papers include D. Grossi [19, 20] who also distinguished between enforcement and regimentation, L. van der Torre, G. Boella and H. Verhagen (see [4] or [5] for example) as well as A. Perreau de Pinninck, C. Sierra and M. Schorlemmer [30], A. Artikis, M. Sergot and J. Pitt [1], M. Esteva, J. Padget and C. Sierra [15] or A. Garcia-Camino, P. Noriega and J. A. Rodriguez-Aguilar [17]. All these papers elaborate on the importance of suitable mechanisms for ensuring compliance in distributed systems and propose mechanisms for specific scenarios. However in these papers, little analysis can be found, examining and comparing the different compliance ideas based on a common setting and researching on the interplay of the different concepts as well as their applicability for certain settings. That is why, this paper aims at providing a first step into the research just mentioned by presenting a comprehensive taxonomy for ensuring compliance with institutions, that can be used not only as a basis for analyzing different compliance mechanisms, but also for comparing, combining and in general developing corresponding strategies.

3 A Taxonomy for Ensuring Institutional Compliance in Utility Computing

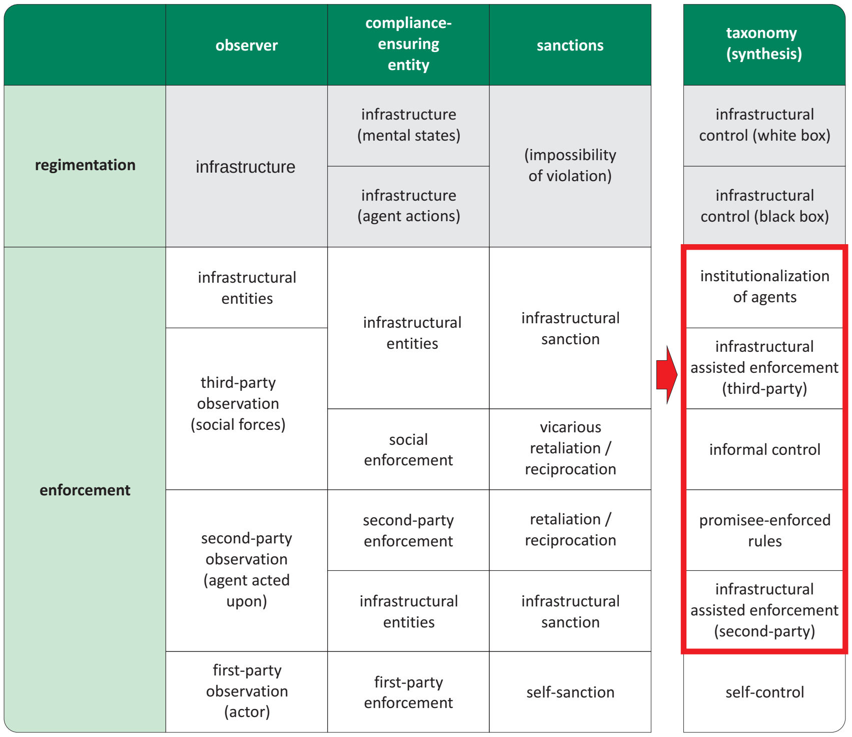
After having had a brief look at the existing literature about institutions and the ensuring of their compliance in the last section, in this section a taxonomy of all methods through which compliance can be administered shall be developed. The goal of this endeavor is to illuminate the general concept of ensuring compliance as well as its different potential forms of implementation.Thereby, first of all, the cornerstones of the taxonomy (the column heads in figure 1) will be explained. This will be followed by a detailed analysis of the resulting compliance-ensuring mechanisms. The ideas for the taxonomy are based on works by Ellickson [13] (that were already cited by North [28] as theoretical “enforcement” foundation) and works by Grossi [19, 20] who made a distinction between regimentation and enforcement and proposed a basic classification mechanism for enforcement concepts.

To start, as already defined in section 2.1, an institutionally tailored system consists of a framework of *rules* defining normatively appropriate behavior. The compliance with these rules is ensured through (positive or negative) *sanctions*, the administration of which is itself governed by rules. Concerning the sanctions, institutionally entailored systems typically employ both rewards and punishment – both carrots and sticks – to influence behavior. In order to administer these positive and negative sanctions, agent behavior is usually divided into three categories [13]:

1. good behavior that is to be rewarded,
2. ordinary behavior that warrants no response (as giving a response to the most common behavior only tends to increase the costs of administering sanctions) and therefore will not be discussed any further in this paper, and
3. negative behavior that is to be punished.

However, before any compliance-ensuring can take place another aspect has to be thought about: the behavior of the agents needs to be monitored in order to categorize it and apply the right kind of sanction[3]. This monitoring can

[3] In the further course of this paper, the main focus will be on sanctions that punish negative behavior, as these are especially relevant in the context of strategic uncer-



	observer	compliance-ensuring entity	sanctions	taxonomy (synthesis)
regimentation	infrastructure	infrastructure (mental states)	(impossibility of violation)	infrastructural control (white box)
		infrastructure (agent actions)		infrastructural control (black box)
enforcement	infrastructural entities	infrastructural entities	infrastructural sanction	institutionalization of agents
	third-party observation (social forces)			infrastructural assisted enforcement (third-party)
		social enforcement	vicarious retaliation / reciprocation	informal control
	second-party observation (agent acted upon)	second-party enforcement	retaliation / reciprocation	promisee-enforced rules
		infrastructural entities	infrastructural sanction	infrastructural assisted enforcement (second-party)
	first-party observation (actor)	first-party enforcement	self-sanction	self-control

Fig. 1. Taxonomy for Ensuring Institutional Compliance

be done by *observers* in a system. Thereby it seems useful to distinguish between 4 types of observers, that not only monitor the behavior of the individual agents, but can act as information source for both rules of behavior and sanctions: a first-party observer who controls his accordance with the rules in a system (whether self-imposed or imposed by other sources) himself, a second-party observer who observes the behavior of his transaction partner(s), third-party observers that control the behavior of other agents the system and last but not least the infrastructure (in the sense of both, the infrastructure as a whole as well as infrastructural entities) as observer. Once, the behavior of the agents is observed, the ensuring of compliance can take place. This can either be done by te observer of the violation or by another party. In total this paper distinguishes 4 different kinds of *compliance-ensuring entities* which all have different kinds of sanctions that can be used for ensuring institutional compliance. The 4 enforcers are: the infrastructure provided by the UtiC designer (including institutional entities as a sub-group), social groups (up to the society as whole) consisting of non-infrastructural entities, second-party enforcers (i.e. the transaction partners) and first party-enforcers.

As a result of these considerations, the *taxonomy* that can be seen in the final column of figure 1 can be developed. The taxonomy is the synthesis of the 4 types of observers that can spot the behavior of agents with regard to the institutional framework (e.g. violations or actions in accordance with the institutions) and the 4 types of compliance ensurers that (depending on their type) can apply sanctions (in order to ensure compliance). It consists of 8 different kinds of combined systems that represent all compliance-ensuring concepts that can be applied reasonably: *infrastructural control (white box)*, *infrastructural control (black box)*, *institutionalization of other agents*, *infrastructural assisted enforcement (third-party)*, *promisee-enforced rules*, *infrastructural assisted enforcement (second-party)* and *self-control*.

3.1 Regimentation vs. Enforcement

After briefly explaining the main categories (i.e. the column heads in figure 1), as a last step before going into detail about the synthesized taxonomy, the distinction between the two row heads of figure 1, i.e. *regimentation* and *enforcement* shall be explained.

Regimentation refers to the ensuring of institutional compliance by making violation states unreachable via an appropriate infrastructure (i.e. allowing for no deviation from institutionally defined behavior) so that no compliance issues occur [23]. This is normally done in either of the following two ways.

1. By ensuring that all agents' mental states are accessible to the system (closed systems), and can be altered to be in accordance with the normative framework. Thus, agents are treated as a white box that's content can be by

tainties. However all the taxonomy elements of this paper can be thought of in form of reward mechanisms for good behavior as well.

analyzed and altered (this concept is for example applied in the KAoS architecture [6]). In the taxonomy this idea is referred to as *infrastructural control (white box).*

2. In case the mental states are not accessible to the system (i.e. the inner states of an agent are a black box to the system), compliance is ensured by constraining the actions of the individual agents. This idea is for example used in systems such as ISLANDER that uses so-called "governors". In ISLANDER agents do not act directly but through their governor, who can consequently check all actions. Hence, if an agent wants to send a message that is not allowed, the governor will not send it and consequently institutional compliance is ensured [14]. In the taxonomy this idea is referred to as *infrastructural control (black box).*

In contrast to regimentation where non-compliance is made impossible by controlling everything that might lead to a violation of the institutional framework, enforcement "only" uses indirect mechanisms in order to ensure compliance. Thus in enforcement positive or negative incentives are being used that shall render compliance the preferable choice for an agent.

Putting it simple: regimentation pursues the idea of 100 per cent control (of either agent actions or their mental states) and consequently compliance can be always be ensured, however it limits the autonomy of the agents. Furthermore it seems difficult to implement it in open distributed settings such as UtiC and might become inoperative in case agents have agreed to conduct the actual transaction outside the monitored environment (of course messages of such type could be filtered by the system, this aspect is neglected at this point). Looking at eBay for example, although the transaction partners agree to live up to their agreements in a transaction (e.g. deliver a good after the money has been set), eBay cannot force them to do so, because the physical transaction takes place outside the eBay marketplace and thus at that point eBay has no direct control over either the mental states or the actions of the individuals acting on eBay. Last but not least one further possible problem arises with regimentation, a problem with its costs. The term costs thereby is not necessarily understood in monetary terms, but can for example be seen in the increased number of messages (infrastructural resources) that are needed for the 100 per cent monitoring. This is where enforcement steps in. Although maybe preferable in some situations, enforcement aims at as much control as possible control at resonable costs for the compliance. As already mentioned it instead makes use of negative and positive incentives that can be applied not only by the organization, but all agents acting in the system as well and therefore can reduce the costs of UtiC designers, by reducing their monitoring work.

Now that the heads of the table columns have been discussed, finally the combined systems that result from the intersection of the components of compliance-ensuring shall be explained in more detail. These are infrastructural control (white / black box), the institutionalization of other agents, infrastructural assisted enforcement (third-party / second-party), informal control, promisee-enforced rules and self control. Thereby special focus will be on the enforcement related concepts and infrastructural control (i.e. regimentation) will be neglected as it has just been discussed.

3.2 Institutionalization of Agents

The institutionalization of other agents can be thought of in form of the implementation of agents with special rights (i.e. some kind of police agents) that patrol the environment (in our example the UtiC environment) and sanction negative behavior (i.e. non-compliance) if spotted. These police agents are given their special rights by the UtiC designer (i.e. they are infrastructural entities and receive their power from the institutional framework provided by the UtiC infrastructure) and consequently perform an institutional compliance-ensuring. However in contrast to regimentation the police agents do not control all actions but only act as enforcers if violations are spotted. The spotting of the institution-violation is done by the police agents themselves who test the behavior of agents at random and react to what they detect. Looking at the kind of sanctions that can be applied by the police agents several sanctions can be thought of (depending on the severity of the non-compliance) such as a complete exclusion of the UtiC system to penalty payments or replacement deliveries of the resources (e.g. disk space).

3.3 Infrastructural Assisted Enforcement (Second-Party / Third-Party)

The concepts of infrastructural assisted enforcement are very close to the idea of the institutionalization of other agents. Thus again infrastructural entities act as compliance ensuring entities that can make use of sanctions ranging from a complete exclusion of the UtiC system to penalty payments or replacement deliveries of the resources (e.g. disk space). However in contrast to the concept of the institutionalization of other agents, not the infrastructural entities act as observers, but either the agent that was acted upon, i.e. the agents that was deceived by its transaction partner (second-party observer) or the observation is done by a third-party, i.e. an agent that is not involved in the transaction but has spotted the non-compliance of one actor. These observers then call the infrastructural entities for conducting the sanctioning in order to assure compliance. Thus, in contrast to the institutionalization of other agents where the infrastructural entities act on their own observations, in these two cases, an additional communication effort must be made that bears two problems. First of all the additional communication needed might result in a longer reaction time and furthermore, the infrastructural entities need to verify the testimonies made to them as the agents my lie on purpose in order to have rival agents sanctioned (and thereby profit themselves).

One sample application of this taxonomy element (with second-party observers) was described by Balke and Eymann [2, 3] that seized an idea by Güth and Ockenfels [22] and analyzed the effects of an arbitration board as infrastructural entity that can be called by any agent that has been deceived. Using an game-theoretic approach, in their paper they showed that with the help of the arbitration board, it is possible to increase trust and reduce strategic uncertainty in open environments such as UtiC markets where software agents trade standardized resources on behalf of their human owners, even if the arbitration board is not equipped with superior detection capabilities, but uses Bayesian rules for assessing the trustworthiness of the agents.

3.4 Informal Control and Promisee-Enforced Rules

Two other concepts that can be thought of where second- or third party observer information is being used are informal control and promisee-enforced rules. Although looking different in figure 1 at the first glance (i.e. in promisee-enforced rules concept it is the agent that has been promised something (i.e. he is a promisee) but didn't receive it as promised who observes and sanctions the non-compliance, whereas in the informal control concept third-party agents observe and sanction) the two concepts are closely interrelated and are therefore presented together in this section. This interrelation can be understood best when thinking about examples for the two concepts. Thus promisee-enforced rules can be found in image-based trust mechanisms, whereas informal control can be found in reputation mechanisms.

Image is a global or averaged evaluation of a given target on the part of an individual. It consists of a set of evaluative beliefs [27] about the characteristics of a target. These evaluative beliefs concern the ability or possibility for the target to fulfill one or more of the evaluator's goals, e.g. to behave responsibly in an economic transaction. An image, basically, tells whether the target is “good” or “bad”, or “not so bad” etc. with respect to a norm, a standard, a skill etc.

In contrast *reputation* is the process and the effect of transmission of a target image. The evaluation circulating as social reputation may concern a subset of the target's characteristics, e.g. its willingness to comply with socially accepted norms and customs [11].

Putting it simple, an image is the picture an individual has gained about someone else (the target) based on his own previous observations of that target. If using reputation, the individual expands the information source about the target beyond its own scope and includes the information of others about the target as well [24].

Applying this to the taxonomy example the following picture can be drawn: in the promisee-enforced rules concept, it is the promisee who acquires an image of the agent it is interacting with. In case the other agent does not perform as promised that promisee can sanction the non-compliance by for example not interacting with the agent once more, etc. In contrast in case of informal control, third-party agents observe a transaction and form an own image about the transaction participant. Then the individual images of agents are shared between the agents and hence they are aggregated by the society (e.g. with the help of gossip) and agents that did not comply with the institutional framework have to fear that every agent that receives the information about their non-compliance will not act with them in the future, thus in this example the whole society functions as enforcers.

3.5 Self-Control

The last part of the taxonomy that shall be discussed in this paper is self-control. In contrast to all other compliance-ensuring mechanisms presented so far, it does not included any additional party, but only the agent performing an action itself. This agent is assumed to have an own normative value system that it was given by its principal and constantly checks whether his actions are in accordance with that own value system and the institutional framework of the UtiC environment (i.e. the agent is its own observer). Thereby it has to be noted that the two normative value systems (i.e. the private one of the agent and the UtiC one) can contradict and needn't necessarily be consistent with one another. Based on the normative value system the agent can then decide to sanction itself. An example of such a self-control scenario in UtiC could be that an vendor of UtiC resources that didn't deliver what he promised (e.g. he promised 1 Tera byte of hard disk space available, but only could provide 0.99 Tera byte) is discontent with his performance (although the buyer might not have complained) and as a result offers his buyer a refund for the money paid.

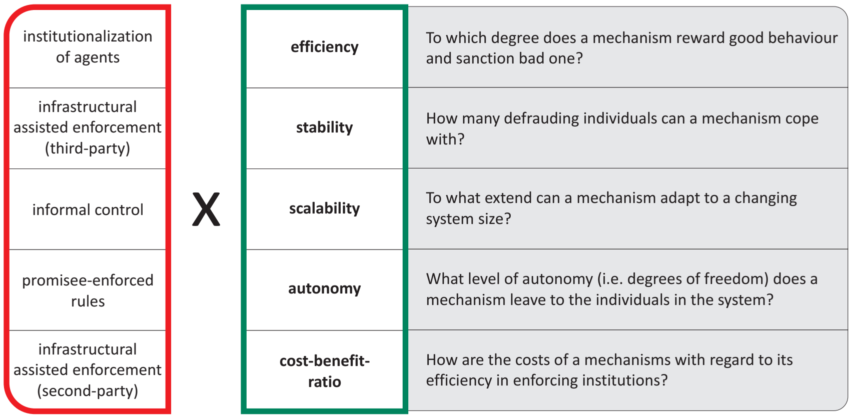
4 Further Research

After presenting this short taxonomy for compliance-ensuring mechanisms in a next step the highlighted enforcement mechanisms shall be evaluated one against the other. Thus, the different enforcement mechanisms will be evaluated against performance indicators derived from literature. These performance indicators can be sen in figure 2.

With the help of the taxonomy developed that aims to prototypically represent existing enforcement mechanisms, an analysis of the technological restrictions of UtiC as well as economic theory, finally a sample UtiC market model without and with the corresponding enforcement mechanisms will be deduced as a next step. This market model will serve as the initial point for the later simulations.

The simulation will be conducted in form of a MAS simulation because MAS offer strong models for representing complex and dynamic environment such as UtiC markets that cannot be analyzed mathematically any more, but need to be simulated. For the simulation a social science simulation research process that is based on works of Gilbert and Troitzsch [18] and Dooley [12] and can be seen in figure 3 will be used.

Looking at the process, first of all an abstract model has to be conceptualized and designed that represents the described UtiC market (with and without the different enforcement mechanisms that are derived from the compliance-ensuring taxonomy) adequately. This includes the consideration of the specification of UtiC. For these UtiC specifications, specifications from existing computational



institutionalization of agents

infrastructural assisted enforcement (third-party)

informal control

promisee-enforced rules

infrastructural assisted enforcement (second-party)

X

efficiency	To which degree does a mechanism reward good behaviour and sanction bad one?
stability	How many defrauding individuals can a mechanism cope with?
scalability	To what extend can a mechanism adapt to a changing system size?
autonomy	What level of autonomy (i.e. degrees of freedom) does a mechanism leave to the individuals in the system?
cost-benefit-ratio	How are the costs of a mechanisms with regard to its efficiency in enforcing institutions?

Fig. 2. Performance Indications vs. Mechanisms

Grid and systems such as the LHC Grid, TeraGrid and GEON-Grid or Grid5000 [35] will be used, as these are existing technical implementations of the economic vision of UtiC.

Once the model has been designed, the building issue needs to be considered, i.e. the model just designed has to be implemented in a MAS simulation environment. Therefore the SimIS simulation environment that is based on the Repast Simphony Simulation Toolkit will be used as it allows to model computational Grid systems in form of "physical" nodes and edges between these nodes, whereas each nodes hosts different agents, which fulfill a certain role each. Afterwards, the next step in the simulation research process is to check if the current model is actually doing what it is expected to do. This process of checking is called verification. In addition to this step the simulation has to be ensured to reflect the behavior of the target, which is called validation. "Validity can be ascertained by comparing the output of the simulation with data collected from the target." [18]

The idea of the simulation experiment is that in the initial form of the simulation, the market model will be implemented in the simulation environment without an enforcement mechanism and will be calibrated in the course of the simulations. Thus, throughout the simulation the UtiC market setting will be altered in terms of the enforcement mechanism applied. In the analysis of the simulation results afterwards, the initial form of the market as well as the market outcome depending on the enforcement mechanism will serve as a reference for the efficiency of different enforcement concepts with regard to the UtiC market setting.

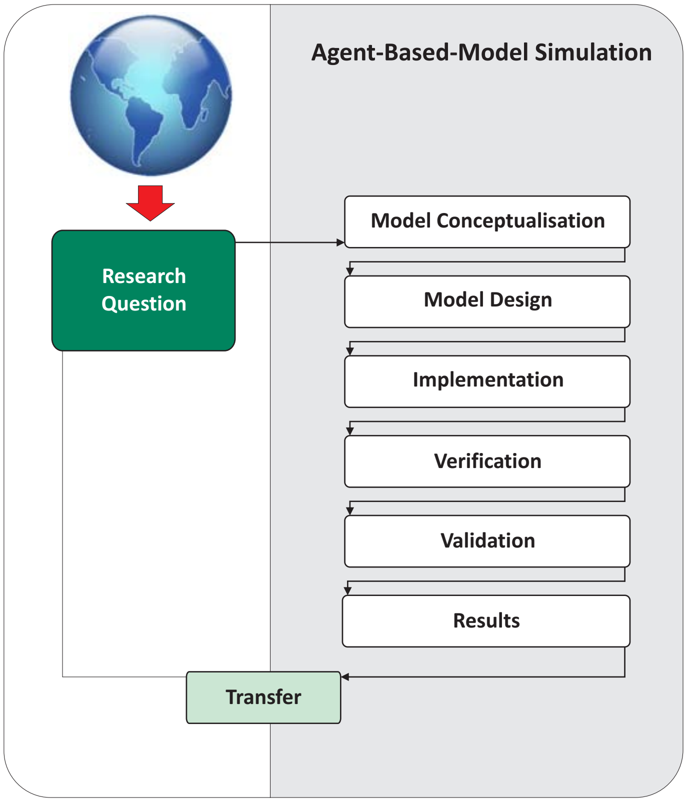


Fig. 3. Simulation Process

After the test of the hypotheses and the corresponding calibration of the simulation, the simulation results will be analyzed and evaluated in order to arrive at set-up specific simulation results in a first step, as well as generalize-able results for the UtiC domain in a second step. Resulting from these evaluations in a last step a generalization is aimed at analyzing which enforcement mechanisms works best for UtiC in which situation.

5 Conclusion

Using an UtiC example, in this paper a taxonomy for ensuring-compliance in open distributed systems was presented. The taxonomy that was synthesized based on considerations of the components and participants of compliance ensuring mechanism in general (see section 3) consists of 8 idealized concepts that were discussed with the help of examples in the further course of the paper. The author views these concepts as a basis for not only analyzing different compliance mechanisms, but also for comparing, combining and in general developing corresponding strategies. Thus in the future work the author plan to simulate prototypical implementations of the taxonomy elements (all based on the same simulation setting) and analyze the performance with regard to compliance ensuring and especially the corresponding cost-benefit ratio. Furthermore, a detailed analysis of the interplay of the taxonomy elements will be made, as in theory not only the individual taxonomy elements are realistic for compliance ensuring strategies, but any combination of the elements is thinkable. However in order to derive at this point, first of all the very high-level concepts presented in this paper need to be made "processable". That means that first off all, in the next step the concepts will be analyzed with regard to their transferability to a logical sound and operational Agent-Based Model. This model will then be used as described in chapter 4. This means that an Agent-Based Model as "processable" model of the economic theory will be developed that will that serve as starting point for a MAS simulation. This simulation aims at evaluating the enforcement concepts that were presented in this paper with regard to the performance indicators mentioned before. With the help of the results the authors hope to be able to draw more general conclusions and arrive at propositions which enforcement concept seem appropriate if only certain performance indicators need to be fulfilled.

References

1. A. Artikis, J. Pitt, and M. Sergot. Animated specifications of computational societies. In *AAMAS '02: Proceedings of the first international joint conference on Autonomous agents and multiagent systems*, pages 1053–1061, New York, NY, USA, 2002. ACM.
2. T. Balke. An extended "court game" - using institutions to foster compliance in open multi-agent systems. In *Proceedings of the Sixth European Workshop on Multi-Agent Systems (EUMAS 2008)*, 2008.

3. T. Balke and T. Eymann. Using institutions to bridge the trust-gap in utility computing markets - an extended "trust game". In *Proceedings of the 9. Internationale Tagung Wirtschaftsinformatik*, 2009.
4. G. Boella and L. van der Torre. A game-theoretic approach to normative multiagent systems. In *Dagstuhl Seminar Proceedings (Dagestuhl Seminar, 18.03. - 23.03.2007)*, Internationales Begegnungs- und Forschungszentrum für Informatik (IBFI), Schloss Dagstuhl, Germany, 2007.
5. G. Boella and L. van der Torre. Substantive and procedural norms in normative multiagent systems. *Journal of Applied Logic*, 6(2):152–171, 2008.
6. J. M. Bradshaw, S. Dutfield, B. Carpenter, R. Jeffers, and T. Robinson. Kaos: A generic agent architecture for aerospace applications. In *Proceedings of the CIKM '95 Workshop on Intelligent Information Agents*, 1995.
7. J. Broersen, M. Dastani, J. Hulstijn, and L. van der Torre. Goal generation in the boid architecture. *Cognitive Science Quarterly*, 2(3-4):428–447, 2002.
8. N. G. Carr. *Does IT matter?* Harvard Business School Press, Boston, MA, 2003.
9. J. S. Coleman. *Foundations of Social Theory.* Harvard University Press, Cambridge, MA, 1990.
10. R. Conte, C. Castelfranchi, and F. Dignum. Autonomous norm-acceptance. In *Intelligent Agents V (ATAL'98)*, volume 155 of *Lecture Notes on Computer Science*, pages 99–112. Springer, Berlin, Germany, 1999.
11. R. Conte and M. Paolucci. *Reputation in Artificial Societies: Social Beliefs for Social Order*. Springer, October 2002.
12. K. Dooley. Simulation research method. In J. Baum, editor, *Companion to Organizations*, pages 829–848. Blackwell, London, 1990.
13. R. C. Ellickson. *Order without Law: How Neighbors Settle Disputes.* Harvard University Press, June 2005.
14. M. Esteva, D. de la Cruz, and C. Sierra. Islander: an electronic institutions editor. In *AAMAS '02: Proceedings of the first international joint conference on Autonomous agents and multiagent systems*, pages 1045–1052, New York, NY, USA, 2002. ACM.
15. M. Esteva, J. A. Padget, and C. Sierra. Formalizing a language for institutions and norms. In *ATAL '01: Revised Papers from the 8th International Workshop on Intelligent Agents VIII*, pages 348–366, London, UK, 2002. Springer-Verlag.
16. I. Foster, N. R. Jennings, and C. Kesselman. Brain meets brawn: Why grid and agents need each other. In *Proceedings of the Third International Joint Conference on Autonomous Agents and Multiagent Systems, 2004. AAMAS 2004*, pages 8–15, 2004.
17. A. Garcia-Camino, P. Noriega, and J. A. Rodriguez-Aguilar. Implementing norms in electronic institutions. In *AAMAS '05: Proceedings of the fourth international joint conference on Autonomous agents and multiagent systems*, pages 667–673, New York, NY, USA, 2005. ACM.
18. N. Gilbert and K. Troitzsch. *Simulation for the Social Scientist.* Open University Press, 2005.
19. D. Grossi. *Designing invisible handcuffs. Formal investigations in institutions and organizations for multi-agent systems.* PhD thesis, Utrecht University, 2007.
20. D. Grossi, H. M. Aldewereld, and F. Dignum. Ubi lex, ibi poena: Designing norm enforcement in e-institutions. In P. Noriega, J. Vázquez-Salceda, G. Boella, O. Boissier, M. Dignum, N. Fornara, and E. Matson, editors, *Coordination, Organizations, Institutions, and Norms in Agent Systems II*, pages 101–114. Springer, 2007.

21. W. Güth and H. Kliemt. Evolutionarily stable co-operative commitments. *Theory and Decision*, 49:197–221, November 2000.
22. W. Güth and A. Ockenfels. The coevolution of morality and legal institutions – an indirect evolutionary approach. *Journal of Institutional Economics*, 1(02):155–174, December 2005.
23. A. J. I. Jones and M. Sergot. On the characterization of law and computer systems: the normative systems perspective. In *Deontic logic in computer science: normative system specification*, pages 275–307. John Wiley and Sons Ltd., Chichester, UK, 1993.
24. S. König, T. Balke, W. Quattrociocchi, M. Paolucci, and T. Eymann. On the effects of reputation in the internet of services. In *Proceedings of the 1st Int. Conference on Reputation (ICORE 2009)*, 2009.
25. M. Luck, P. McBurney, O. Shehory, and S. Willmott. *Agent Technology: Computing as Interaction (A Roadmap for Agent Based Computing)*. AgentLink, 2005.
26. J. Martiensen. *Institutionenökonomik*. Verlag Vahlen, 2000.
27. M. Miceli and C. Castelfranchi. The role of evaluation in cognition and social interaction. In K. Dautenhahn, editor, *Human cognition and social agent technology*. Benjamins, Amsterdam, 2000.
28. D. C. North. *Institutions, Institutional Change and Economic Performance (Political Economy of Institutions and Decisions)*. Cambridge University Press, October 1990.
29. D. C. North. Institutions matter. Economics history working paper, 1994.
30. A. Perreau de Pinninck, C. Sierra, and M. Schorlemmer. Friends no more: norm enforcement in multiagent systems. In E. H. Durfee and M. Yokoo, editors, *AAMAS '07: Proceedings of the 6th international joint conference on Autonomous agents and multiagent systems*, pages 640–642, New York, NY, USA, 2007. ACM.
31. L. Rasmusson and S. Janson. Agents, self-interest and electronic markets. *The Knowledge Engineering Review*, 14(2):143–150, 1999.
32. J. Sabater. *Trust and reputation for agent societies*. PhD thesis, Institut d'Investigació en Intelligència Artificial, Universitat Autonòma de Barcelona, 2003.
33. A. Schotter. The evolution of rules. In R. N. Langlois, editor, *Economic as process*, pages 117–133. Cambridge University Press, Cambridge, 1986.
34. C. Sierra and P. Noriega. Agent-mediated interaction. from auctions to negotiation and argumentation. In *Selected papers from the UKMAS Workshop on Foundations and Applications of Multi-Agent Systems*, pages 27–48, 2002.
35. W. Streitberger. *Einsatz von Risikomanagement bei der Steuerung von Grid-Systemen - Ein Analyse von Versicherungen anhand einer simulierten Grid-Ökonomie*. PhD thesis, University of Bayreuth, 2009.
36. J. Vázquez-Salceda, H. Aldewereld, and F. Dignum. Implementing norms in multiagent systems. In *Multiagent System Technologies*, volume 3187 of *LNAI*, pages 313–327. Springer, 2004.
37. H. Verhagen. *Norm Autonomous Agents*. PhD thesis, Stockholm University, 2000.
38. S. Voigt. *Institutionenökonomie*. Neue Ökonomische Bibliothek. UBT Verlag, 2002.

An essay on *msic*-systems

Jan Odelstad

1) Department of Mathematics, Natural and Computer Sciences, University of Gävle, Sweden, 2) DSV, KTH, Sweden, jod@hig.se

Abstract. A theory of many-sorted implicative conceptual systems (abbreviated *msic*-systems) is outlined. Examples of *msic*-systems include legal systems, normative systems, systems of rules and instructions, and systems expressing policies and various kinds of scientific theories. In computer science, *msic*-systems can be used in, for instance, legal information systems, decision support systems, and multi-agent systems. In this essay, *msic*-systems are approached from a logical and algebraic perspective aiming at clarifying their structure and developing effective methods for representing them. Of special interest are the most narrow links or joinings between different strata in a system, that is between subsystems of different sorts of concepts, and the intermediate concepts intervening between such strata. Special emphasis is put on normative systems, and the role that intermediate concepts play in such systems, with an eye on knowledge representation issues. In this essay, normative concepts are constructed out of descriptive concepts using operators based on the Kanger-Lindahl theory of normative positions. An abstract architecture for a norm-regulated multi-agent system is suggested, containing a scheme for how normative positions will restrict the set of actions that the agents are permitted to choose from.

Key-words: Concept formation, Intermediary, Intermediate concept, Legal concept, Normative system, Normative position, Norm-regulated system, Agent architecture.

1 Introduction

1.1 Conceptual systems in computer- and systems sciences

In the famous Schilpp-volume where established scholars discuss Einstein's work in physics and philosophy, Einstein, in his reply to criticisms, states the following about the relationship between epistemology and science:

> The reciprocal relationship of epistemology and science is of noteworthy kind. They are dependent upon each other. Epistemology without contact with science becomes an empty scheme. Science without epistemology is—insofar it is thinkable at all—primitive and muddled. However, no sooner has the epistemologist, who is seeking a clear system, fought his way through to such a system, than he is inclined to interpret the thought-content of science in the sense of his system and to reject

> whatever does not fit into his system. The scientist, however, cannot afford to carry his striving for epistemological systematic that far. He accepts gratefully the epistemological conceptual analysis; but the external conditions, which are set for him by the facts of experience, do not permit him to let himself be too much restricted in the construction of his conceptual world by the adherence to an epistemological system. He therefore must appear to the systematic epistemologist as a type of unscrupulous opportunist ... (Einstein, 1949.)

The science Einstein has in mind is primarily physics, but even for sciences that are rather unlike physics its reciprocal relationship to epistemology is of a noteworthy kind. The external conditions that, according to Einstein, restrict the adherence to an epistemological system is "the facts of experience", with what Einstein probably meant the results of observations and experiments. But for the sciences that are rather unlike physics "the facts of experience" may better be characterized in some other way. For computer- and systems sciences, "the facts of experience" may perhaps be described as "useful applications".

Every science ought to critically question its foundational assumptions. How urgent the researchers in a field experience these foundational questions may vary greatly from time to time. But probably all sciences go through stages when the need for revisions and elaborations of the basic principles and fundamental conceptions seem inevitable. In a young science, the foundational problems are important and at the same time not seldom overlooked, since researchers working in the field are so enthusiastic over the flow of new results. In such situations, philosophy (which includes epistemology as one of its sub-disciplines) may have a role to play to make clear—and sometimes even to remedy—weak points in the base of the new discipline. In this essay, some problems in the foundations of computer- and systems sciences are addressed and theories and tools which could be useful in the further development of some aspects of this discipline are outlined.[1]

Concepts are a fundamental tool for all kinds of human communication and concept formation is an important process in all branches of science. Information science is of course not an exception. An information system is, when all technical "embeddings" have been stripped off, a set of concepts and relations between these concepts. The skeleton of an information system is a conceptual structure, and this structure must have a solid formal representation, otherwise it cannot function in a computer context.

The formal representations of conceptual systems has a long history in philosophy and in several scientific disciplines. This essay is focused on the relation between layers or strata of concepts of different sorts in a conceptual system and on intermediate concepts that function as links between different strata. This study is brought about using algebraic tools, which implies that the representation is algebraic in character. The result is a theory of many-sorted implicative conceptual systems, *msic*-systems.

[1] This essay is a revised version of Odelstad (2008b).

I argue for an *anti-nivelistic* approach to theoretical systems, which implies the recognition of the multitude of layers or strata that usually are parts of such systems.[2] As a consequence, I also argue for an anti-nivelistic approach to knowledge representation. The following sketch is very vague and metaphorical, however, my message is more adequately found in the formalism below. Suppose that an *msic*-system M represents knowledge or information of a domain D. The implicative relation between concepts represents knowledge of some kind and the kind of knowledge it represents may differ in different parts of the system. In some parts of the system, it may represent conceptual knowledge, the knowledge of definitions of concepts and the logical relations between concepts. In other parts of the system, it may represent for example empirical knowledge about some kind of phenomena and in yet another part of the system it may represent empirical knowledge of another kind. Different strata of concepts of different sorts may thus express knowledge of different kinds. The knowledge represented by links between different strata often represent knowledge of a kind still different from the knowledge represented by the strata, for example knowledge of rational actions or appropriate rules. The revision of an *msic*-system can be done very partially. In many cases, the necessary revision is effected by the modification of the narrowest links between some strata of different kinds.

It is often argued that, for example, rule-based expert systems cannot be modified by the expert system itself. The following quotation from a text book may illustrate this idea:

> Knowledge in a rule-based expert system is represented by IF-THEN production rules collected by observing or interviewing human experts. This task, called knowledge acquisition, is difficult and expensive. In addition, once the rules are stored in the knowledge base, they cannot be modified by the expert system itself. Expert systems cannot learn from experience or adapt to new environments. Only a human can manually modify the knowledge base by adding, changing or deleting some rules. (Negnevitsky, 2005, p. 261.)

One of the advantages with the anti-nivelistic approach to knowledge representation expressed by *msic*-systems is, as I see it, that this may not be true. This is discussed in connection with forest cleaning below.

1.2 Stratification of concepts in theoretical systems

In an article from 1936, Albert Einstein discusses, among other things, the stratification of the scientific system. According to Einstein, there is a multitude of different layers or strata of concepts in science, where higher layers are more abstract than lower layers. As regards to the final aim of science, Einstein suggests, intermediary layers are only of temporal nature and must eventually disappear

[2] *Nivelistic* is constructed out of the French verb *niveler*, meaning “Mettre au même niveau, rendre égal".

as irrelevant. But in the science of today, these strata represent partial success, though problematic. (See Einstein, 1973, p. 295.)

Many theoretical systems show the same kind of phenomena as theoretical physics in the following respects: In the system there is a hierarchical ordering of the concepts in different strata and the status of the concepts in intermediate strata is not obvious. In theoretical physics, the ordering of the layers is based on degrees of abstraction. In other contexts, the stratification of the system can be grounded on quite different principles, for example: descriptive versus normative, state versus action or physical versus mental. One of the main issues to be examined in this essay is the stratification of concepts in theoretical systems, especially the connections between different strata and the function and status of intermediate layers.

The kind of theoretical systems that will come into focus in this study can, in a fairly general way, be characterized as *conceptual systems* and two essential characteristics of these systems are the following: They have an implicative form and they are many-sorted, i.e. a system consists of different sorts of concepts (at least two). They are thus *many-sorted implicative conceptual systems*, in the sequel abbreviated *msic*-systems. Different kinds of systems belong to the class under study, for example legal systems, normative systems, systems of rules and instructions, systems expressing policies and some varieties of scientific theories. Such systems have an important role to play in the discipline artificial intelligence, which has as one of its aims to bring forth “smart” behaviour of computers.

In the investigation reported here, *msic*-systems are studied from a logical and algebraic perspective aiming at clarifying their structure and developing effective methods for representing them. Special emphasis is put on the most “narrow” links between subsystems of different sorts in a system and intermediaries (intermediate concepts) mediating or intervening between subsystems of different sorts. Such links and intermediaries are of great interest when there are reasons for changing the system.

In computer science, *msic*-systems can be useful in many problem areas, for example: legal information systems, computer security, knowledge representation, expert systems, architectures for multiagent-systems, decision-analytic support systems and agent-based simulations. This study of *msic*-systems is mainly a contribution to the tradition of constructing intelligible and explicit models and representations in contrast to case-based, connectivist and emergent approaches (cf. Luger, 2002, p. 228). But *msic*-systems also prepare the grounds for the use of machine learning, where the links and intermediaries between subsystems will play an important role.

1.3 The theory of *msic*-systems

When developing a theory of *msic*-systems, it is important to note that different parts of the theory are situated on different levels of abstraction, and as a consequence there are different levels of applications of the theory. The word ‘theory’ has several meanings and in this context it is important to distinguish between the following two meanings:

(1) Theory in the sense often used in logic; abstract theory, theory in contrast to model (in the model-theoretic sense)
(2) Theory in contrast to practice and application.

Here a theory of *msic*-systems is put forward in both senses of 'theory'. The theory of *msic*-systems, where 'theory' is taken in the second sense contains some theories of *msic*-systems in the first sense, of formal theories. The formal theories of *msic*-systems are characterized axiomatically as algebraic theories and among the models of these abstract (formal) theories are specific *msic*-systems. The abstract theories of *msic*-systems express the structure of such systems. The theory of *msic*-systems, in the second sense, contains other theoretical perspectives than the abstract, formal ones.

The theory of *msic*-systems will be abbreviated *msic*-theory, where 'theory' stands for sense (2). A formal, abstract theory of *msic*-systems where 'theory' is taken in sense (1) will be called a structural *msic*-theory, since such a theory characterizes the structure of *msic*-systems. Such abstract theories will usually be presented as axiomatized theories within set theory. The most abstract part of the *msic*-theory will be framed as a number of set-theoretical predicates.

2 Normative systems

2.1 What is a norm?

The theory of *msic*-systems has many applications and there are many different kinds of *msic*-systems. In this essay I will focus on the representations of normative systems as *msic*-systems. I take a first step in the analysis of norms in this section, and a great deal of simplification is needed. Modifications and elaborations of this oversimplified picture will be developed step by step in later sections.

Norms, normative sentences, are understood in contrast to descriptive sentences. Sentences of the latter kind express matters of fact but are not used for expressing evaluations or value judgments. A normative sentence, on the other hand, does not state what *is* the case but what *shall* be the case or what *may* be the case, or will have an *evaluating* function.

Let us preliminarily say that there are two kinds of normative sentences, viz. categorically normative sentences and conditional normative sentences. A categorically normative sentence consists of a descriptive sentence preceded by a 'norm creating operator', for example 'it shall be the case that' or 'it may be the case that'. If q is a descriptive sentence then 'it shall be the case that q', which is abbreviated Shall(q) and 'it may be the case that q', abbreviated as May(q), are examples of categorically normative sentences. A conditional norm is an *if-then* sentence (an implication) where the antecedent is descriptive and the consequent is purely normative. Hence, a conditional norm has the form

$$p \to B(q)$$

where p and q are descriptive sentences and B is a norm-creating operator, for example Shall or May. As suggested above, it is possible to extend ordinary propositional logic with propositional operators as Shall and May, etc. The branch of logic derived in this way is called deontic logic. ‘Deontic’ comes from the Greek word ‘deont’, which means “that which is binding”. Expressed in a very general way, deontic logic is the logical study of obligation and permission. The modern study of this kind of logic is often said to have commenced with the article “Deontic Logic” by the Finnish philosopher Georg Henrik von Wright published in *Mind* in 1951.[3] This theory was anticipated by Ernst Mally in the 1920s and, much earlier, by Gottfried Wilhelm Leibniz (1646-1716) and Jeremy Bentham (1748-1832). The core of standard deontic logic is the formal study of the deontic operators ‘it is permissible that’ (May) and ‘it is obligatory that’ (Shall) and we can extend predicate logic as well as propositional logic with these operators.

2.2 Norms in predicate logic and as ordered pairs

A conditional norm is (usually) expressed as a universal sentence. For example:

(n_1) For any x, y and z : if x has promised to pay \$$y$ to z, then x has an obligation to pay \$$y$ to z.

Within predicate logic, we can formalize (n_1) as follows:

(n_2) $\forall x, y, z : \mathit{PromisedPay}(x, y, z) \to \mathit{Obligation\_to\_Pay}(x, y, z)$

Thus, a typical conditional norm is a universal implication. Syntactically it consists of three parts: the sequence of universal quantifiers, the antecedent formula and the consequent formula. Note that the norm (n_2) correlates open sentences: $\mathit{PromisedPay}(x, y, z)$ is correlated to $\mathit{Obligation\_to\_Pay}(x, y, z)$. A norm like (n_2) can therefore be represented as a relational statement correlating a *ground, PromisedPay*, to a *consequence, Obligation_to_Pay*:

$$\mathit{PromisedPay}\ \mathcal{R}\ \mathit{Obligation\_to\_Pay}.$$

Generally, $p\mathcal{R}q$ represents the norm

(n_3) $\forall x_1, ..., x_\nu : p(x_1, ..., x_\nu) \to q(x_1, ..., x_\nu)$

given that p and q are ν-ary predicates. It is important here that the free variables in $p(x_1, ..., x_\nu)$ are the same and in the same order as the free variables in $q(x_1, ..., x_\nu)$. $\mathcal{R}$ is a binary relation, and $p\mathcal{R}q$ is a relational statement equivalent to $\langle p, q\rangle \in \mathcal{R}$. Thus, a norm can be represented as $p\mathcal{R}q$ or $\langle p, q\rangle \in \mathcal{R}$. If, in the

[3] The development of deontic logic is closely related to another, better known part of logic, namely modal logic. The core of modal logic is the formal study of the operators ‘it is possible that’ and ‘it is necessary that’ (the so-called alethic modalities) and modal propositional logic is propositional logic extended with the possibility- and necessity-operator.

actual context, $\mathcal{R}$ can be tacitly understood and therefore omitted, it is only a small step to the representation of (n_3) as the ordered pair $\langle p, q\rangle$.

Note that $p\mathcal{R}q$ as a representation of (n_3) does not generally presuppose that q is a normative (or deontic) predicate, so $p\mathcal{R}q$ can be used as a representation of any sentence which has the same form as (n_3). Therefore, in many contexts of application the implicative relation $\mathcal{R}$ can be such that only some of the sentences $p\mathcal{R}q$ are norms. For reasons that will be explained when the formal framework is discussed, $p\mathcal{R}q$ will be abbreviated as the ordered pair $\langle p, q\rangle$ only when p and q are conditions of different sorts.

In the above discussion of the representation of norms, p and q, as well as $PromisedPay$ and $Obligation\_to\_Pay$, appear as predicates. But the term predicate is often used for syntactical entities, and, therefore, interpreting $p\mathcal{R}q$, p and q will here instead be conceived of as *conditions*. If p is a ν-ary condition and i_1,...,i_ν are individuals, then $p(i_1, ..., i_\nu)$ is a statement. Antecedents and consequences of norms are represented as conditions and are called *grounds* and *consequences* respectively. A norm is represented as a statement relating (or correlating) a ground to a consequence, or represented as an ordered pair consisting of a ground and a consequence. In the preliminary analysis put forward in this section, grounds are descriptive and consequences are normative conditions.[4]

Note that $Obligation\_to\_Pay$ is a normative condition but that the sentence $Obligation\_to\_Pay(x, y, z)$ can be analysed as

Obligatory $Pay(x, y, z)$.

where Obligatory is a deontic operator resulting in a new predicate when it is applied to a given predicate. Pay is a descriptive condition and by applying the deontic operator Obligatory we can in a sense construct a normative condition Obligatory Pay out of the descriptive condition Pay. It is presupposed here that 'Obligatory Pay' is equivalent to $Obligation\_to\_Pay$ and I will return to this way of constructing normative conditions out of descriptive conditions using deontic operators.

Within the framework of the above preliminary analysis of norms, we can view a normative system $\mathcal{N}$ as consisting of a system $\mathcal{B}_1$ of potential *grounds* (descriptive conditions) and a system $\mathcal{B}_2$ of potential *consequences* (normative conditions). The set of norms in $\mathcal{N}$ are the set J of *links* or *joinings* from $\mathcal{B}_1$ to $\mathcal{B}_2$. The Figure 1.1 is an attempt to illustrate the situation, where a norm is represented by an arrow from the system of grounds to the system of consequences.

A norm in a normative system $\mathcal{N}$, the norm here represented as an ordered pair $\langle p, q\rangle$, can be regarded as a mechanism of inference. We can distinguish two cases. Suppose that p and s are descriptive conditions and q and t normative. Then the following "derivation schemata" are valid given $\mathcal{N}$.

1.

$p(i_1, ..., i_\nu)$

$\langle p, q\rangle$

[4] Cf. Odelstad & Lindahl (2002), pp. 32 ff and Lindahl & Odelstad (2004), section 3.2.

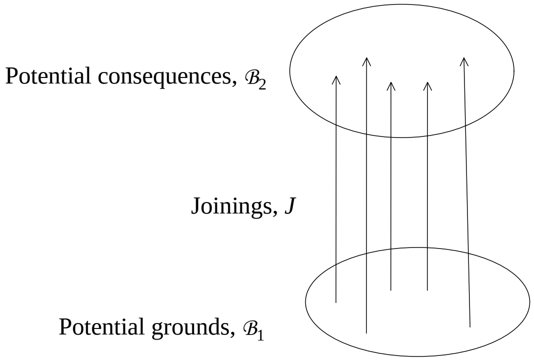

Potential consequences, $\mathcal{B}_2$
Joinings, J
Potential grounds, $\mathcal{B}_1$

Fig. 1. A simple normative system.

$q(i_1, ..., i_\nu)$

2.
$s\mathcal{R}p$
$\langle p, q \rangle$
$q\mathcal{R}t$

$\langle s, t \rangle$[5]

In (1), $\langle p, q \rangle$ functions as a deductive mechanism correlating sentences by means of instantiation, while in (2), $\langle p, q \rangle$ plays an important role in correlating one condition, s, to another condition, t.[6]

A condition, as the term is used here, is very similar to a relation; in a sense a condition is used for "expressing" a relation.[7] Relations, and therefore also conditions, are a specific kind of concepts. A normative system is thus a system consisting of an implicative relation between concepts. Note that the kind of normative systems we have encountered so far consists of two sorts of concepts, descriptive and normative.

[5] Note that $s\mathcal{R}p$ relates conditions of the same sort and the same holds for $q\mathcal{R}t$; s and p are descriptive but q and t are normative. A norm consists of conditions of different sorts. As stated earlier, only implicative sentences that relate conditions of different sorts will be represented as ordered pairs.

[6] See Lindahl & Odelstad (2004) subsection 3.2 and Odelstad & Boman (2004) subsection 2.2. Cf. Alchourrón & Bulygin (1971) p. 28. Schema 1 corresponds to what Alchourrón and Bulygin call the correlation of individual cases to individual solutions, and schema 2 corresponds to what they call the correlation of generic cases to generic solutions.

[7] Properties are here regarded as unary relations and can be "expressed" by conditions.

Easily observable, conjunctions, disjunctions and negations of conditions can be formed by the operations $\wedge, \vee, '$, namely in the following way (where $x_1, ..., x_\nu$ are place-holders, not individual constants).

$(p \wedge q)(x_1, ..., x_\nu)$ if and only if $p(x_1, ..., x_\nu)$ and $q(x_1, ..., x_\nu)$.
$(p \vee q)(x_1, ..., x_\nu)$ if and only if $p(x_1, ..., x_\nu)$ or $q(x_1, ..., x_\nu)$.
$(p')(x_1, ..., x_\nu)$ if and only if not $p(x_1, ..., x_\nu)$.

$\perp$ (Falsum) is the empty condition, not fulfilled by any $\nu-$tuple, and $\top$ (Verum) is the universal condition, fulfilled by all $\nu-$tuples.

As is well-known, the truth-functional connectives can be used as operations in Boolean algebras. It is therefore possible to construct Boolean algebras of conditions. The role of the set of norms is to join two Boolean algebras:

- a Boolean algebra of grounds,
- a Boolean algebra of consequences.

The norms are links or joinings between the algebra of grounds and the algebra of consequences.

The outline of the algebraic approach to normative systems just presented is substantially simplified. The approach will be developed extensively below.

3 Conceptual systems

In the previous subsection, a simple normative system has been characterized as a two-sorted implicative conceptual system, where the concepts are, from a logical point of view, relations (expressed as conditions) and the two sorts of concepts involved are descriptive and normative conditions. However, relations (and therefore also conditions) are only one specific kind of concepts, where 'kind' is something else than 'sort'. Other kinds of concepts are, *inter alia*, aspects (in philosophy of science often called attributes) and measures (often termed scales). Examples of aspects are length, weight, temperature, intelligence, utility and probability. Examples of measures are meter, kilogram, degrees centigrade and the probability measure. Different *kinds* of concepts have different logical form (for example relations, structures and functions) while different *sorts* of concepts differ in their cognitive status (for example descriptive and normative respectively). From a logical point of view, aspects are structures and measures are functions.

When studying implicative conceptual systems where the concepts are conditions, the implicative relation is implication in a straightforward sense. However, when the concepts are aspects or scales, we are dealing with implicative relations that are implications only in a rather generalized sense. Implicative statements, i.e. statements expressing that an implicative relation holds, can in such cases, for example, be interpreted as determination or relevance. However, we will even in the generalized contexts talk about the antecedent and consequent of an implicative statement, and even of grounds and consequences.

As pointed out above, for concepts which are conditions we can in an obvious way define the operations conjunction, disjunction and negation and thereby arrive at a Boolean algebra. The situation is different for concepts that are aspects, since taking the negation of an aspect is not certainly a meaningful operation. However, aspects can form a lattice. We shall discuss this further below.

'Concept' is a complicated notion and is of great importance in many areas. It is tightly connected to the notion of 'meaning', and 'the meaning of concepts' is a philosophical minefield. But in this context, it is impossible to avoid the term 'concept'. The following short passage from the entry *Concept* in *The Encyclopedia of Philosophy* describes its usefulness:

> Concept is one of the oldest terms in the philosophical vocabulary, and one of the most equivocal. Though a frequent source of confusion and controversy, it remains useful, precisely because of its ambiguity, as a sort of passkey through the labyrinths represented by the theory of meaning, the theory of thinking, and the theory of being. (Heath, 1967.)

In the theory of *msic*-systems, the use of the notions 'concept' and 'meaning' is instrumental, and these notions function as passkeys to the main objectives of the work presented here. As the word 'concept' is used, complex combinations of concepts are still regarded as concepts. A concept can be defined in terms of other concepts in a more or less complicated way.

A notion connected to 'concept' that will play a role here is 'cognitive status'. The idea is that the different sorts of concepts constituting an *msic*-system are often different with respect to their cognitive status. As a source of inspiration for using the notion 'cognitive status' in the theory of *msic*-systems one can take Ernest Nagel's discussion of the cognitive status of scientific theories in his book *The Structure of Science*. But here the term 'cognitive status' is applied to concepts. Examples of different cognitive status include: logical, empirical, observational, operational, theoretical, physical, mental, descriptive, prescriptive, normative, evaluative, and—as we will see below—intermediate. (Note that several of the different sorts of cognitive status exemplified above can be applied to the same concepts.)

4 Intermediate concepts—form and function

4.1 Intermediaries

In the simplified presentation above, a normative system is represented as a two-sorted implicative conceptual system, consisting of a set of descriptive grounds and a set of normative consequences. However, many concepts for example in law are neither purely descriptive nor purely normative. Like Janus, the Roman god of beginnings and endings, they have two faces, one turned towards facts and description, the other towards legal consequences. These concepts are said

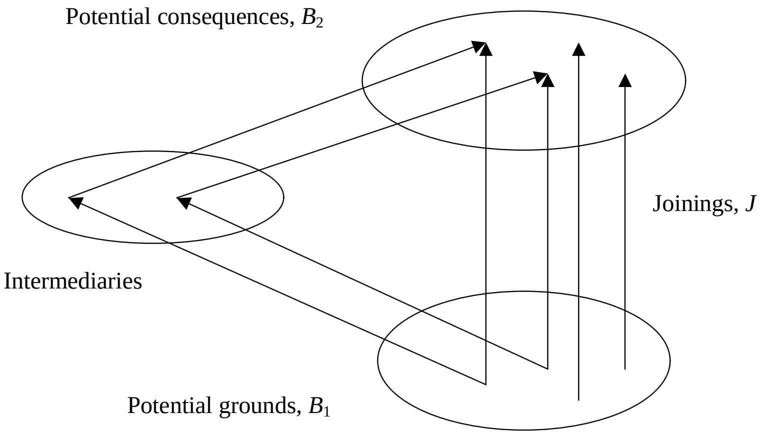


Fig. 2. A normative system with intermediaries.

to be intermediate between facts and legal consequences and will often be called intermediaries. Figure 1.2 will give a first illustration of this idea.

As an example, consider what it means to be a citizen according to the system of the U.S. Constitution. Article XIV, section 1 reads as follows:

> All persons born or naturalized in the United States, and subject to the jurisdiction thereof, are citizens of the United States and of the State wherein they reside. No State shall make or enforce any law which shall abridge the privileges or immunities of citizens of the United States; nor shall any State deprive any person of life, liberty, or property, without the due process of law; nor deny to any person within its jurisdiction the equal protection of the laws.

Two key concepts in the article are *citizen* and *person.* The article specifies the ground for the condition being a citizen in the United States:

persons born or naturalized in the United States, and subject to the jurisdiction thereof

and specifies a number of regal consequences of this condition expressed in terms of ‘shall’:

no State shall make or enforce any law which shall abridge the privileges or immunities of citizens of the United States.

The article does not state any ground for the condition to be a person but specifies a number of legal consequences connected to this condition:

nor shall any State deprive any person of life, liberty, or property, without due process of law; nor deny to any person within its jurisdiction the equal protection of the laws.

Within the constitutional system of United States, this article is supplemented with rules laid down by the Constitution and through court decisions. These rules determine together, by specifying grounds and consequences, the role the concept 'citizen' and 'person' have within the legal system.

Let us construct a simplified "condition-implicative" representation of the legal rules described above.[8] According to the rules, the disjunction of the two conditions

b: to be a person born in the U.S.

n: to be a person naturalized in the U.S.

in conjunction with the condition

s: to be a person subject to the jurisdiction of the U.S.

implies the condition

c: to be a citizen of the U.S.

That this implicative relationship holds according to the system is represented in the form $((b \vee n) \wedge s)\mathcal{R}c$. Since it is a settled matter that citizens who are minors do not have the right to vote in general elections, c does not imply the condition

e: to be entitled to vote in general elections.

Therefore: not $[c\mathcal{R}e]$, and hence not $[((b \vee n) \wedge s)\mathcal{R}e]$.

Let

a: to be adult.

Simplifying matters, suppose that,

(1) $(c \wedge a)\mathcal{R}e$.

It is easy to see that this is equivalent to

(2) $c\mathcal{R}(a' \vee e)$.

Going from (1) to (2) can be called *exportation*, and going from (2) to (1) *importation*.

We thus have within the system the following rules: $((b \vee n) \wedge s)\mathcal{R}c$ and $c\mathcal{R}(a' \vee e)$, stating that the condition $((b \vee n) \wedge s)$ is a ground for c and $(a' \vee e)$ is a consequence of c. These two rules determine partly the role of c (citizenship) in the constitutional system under study. But there can also be other grounds $g_1, g_2, ...$ for c and consequences $h_1, h_2, ...$ of c within the constitutional system. Suppose that $g_1, g_2, ...$ are the grounds of c and $h_1, h_2, ...$ the consequences of c. Hence, the role of c in the system is characterized by

$$g_1\mathcal{R}c, g_2\mathcal{R}c, ..., c\mathcal{R}h_1, c\mathcal{R}h_2, ...$$

The concept c thus couples a set of legal consequences to a set of legal grounds and c is situated "intermediate" between the set of grounds and the set of consequences. Concepts of this kind are called *intermediate concepts* or *intermediaries*. Over the past sixty years, there has been an on-going discussion in Scandinavia as regards the idea of intermediate concepts in the law. The debate was started

[8] The concept citizen regarded as an intermediary is discussed in Odelstad & Lindahl (1998), Odelstad & Lindahl (2000) and Lindahl & Odelstad (2000). In Lindahl & Odelstad (2003), citizenship is treated from the point of view of organic wholes.

in 1944-1945 by Anders Wedberg and Per-Olof Ekelöf, and in 1951 Alf Ross published his well-known essay on ”Tû-Tû”.[9] In this debate, an often used example is the concept of ownership. Ross represents a set of legal rules concerning ownership (denoted O) in essentially the following way, where F_i expresses a possible legal ground and C_j a legal consequence of O.

$$\left.\begin{array}{l} F_1 \text{ ®} \\ F_2 \text{ ®} \\ F_3 \text{ ®} \\ \vdots \\ F_p \text{ ®} \end{array}\right\} \quad O \text{ ®} \quad \left\{\begin{array}{l} C_1 \\ C_2 \\ C_3 \\ \vdots \\ C_n \end{array}\right.$$

Ross himself comments on this scheme in the following way:

> “O” (ownership) merely stands for the systematic connection that F_1 as well as F_2, F_3,...,F_p entail the totality of legal consequences C_1, C_2, C_3,..,C_n. As a technique of presentation this is expressed then by stating in one series of rules the facts that “create ownership” and in another series the legal consequences that “ownership” entails. (Ross 1956-57, p. 820.)

Note that the rules that “create ownership” can be expressed by one rule: $F_1 \vee ... \vee F_p \longrightarrow O$.[10] And the rules describing what ‘ownership’ entail can also be condensed to one rule: $O \longrightarrow C_1 \wedge ... \wedge C_n$. So an equivalent way of representing the legal rules concerning ownership according to Ross is the following scheme:

$$F_1 \vee ... \vee F_p \longrightarrow O \longrightarrow C_1 \wedge ... \wedge C_n$$

Whereas $F_1, ..., F_p$ can be called grounds and $C_1, ..., C_n$ consequences of O, $F_1 \vee ... \vee F_p$ is *the* ground of O and $C_1 \wedge ... \wedge C_n$ *the* consequence of O.

Note that the rule $F_i \rightarrow O$ is a way of introducing O into the discourse, and appropriately we can call such a rule an *introduction rule* of O. In harmony with this, the rule $O \longrightarrow C_j$ can be called an *elimination rule* of O, since in a sense such a rule can eliminate O from the discourse. Analogous to the use of the phrases ’the ground’ and ’the consequence’ we can say that

$$F_1 \vee ... \vee F_p \longrightarrow O$$

is *the* introduction rule of O and

$$O \longrightarrow C_1 \wedge ... \wedge C_n$$

[9] For a more detailed analysis of the early Scandinavian debate see Lindahl & Odelstad (1999a) section 1.2.

[10] $\longrightarrow$ is a consequence relation.

is *the* elimination rule of O.[11]

In Wedberg (1951), three different methods for treating the concept of 'ownership' are discussed. The first and second of these methods aim at a definition of ownership in terms of grounds and consequences respectively. Wedberg's third method treats ownership as a 'vehicle of inference'. According to Wedberg this means that ownership is a tool for inferring statements of legal consequences from statements of legal facts, and, therefore, ownership is undefined. Obviously, Wedberg's third method for treating ownership is close to Ross's view.

We will return to the question of defining intermediate concepts in relation to regarding them as vehicles of inferences. As a point of departure for further discussions and refinements, we regard intermediate concepts as characterized by their grounds and consequences. The characterization of the concept citizenship, c, thus has the following form:

$$g_1\mathcal{R}c, g_2\mathcal{R}c, ..., c\mathcal{R}h_1, c\mathcal{R}h_2, ...$$

For the view of intermediate concepts adopted in this essay, the discussion in legal philosophy has been an important source of inspiration. But there are of course also other theories that have influenced this research. The following quotation from Lindahl & Odelstad (1999a) emphasizes this, where "the ideas mentioned above" are the ideas of Wedberg and Ross.

> In the theory of language of Michael Dummett, there are features with some resemblance to the ideas mentioned above. According to Dummett, the meaning of an expression is determined, on one hand by the condition for correctly uttering it, and on the other hand by what the uttering of the expression commits the speaker to. Therefore, the meaning of a statement is identified in part by the conditions from which it can be inferred and in part by what can be inferred from the statement. In the case of utterances of sentences composed by the connectives "and", "or" etc., this is given by what are called introduction and elimination rules in Gentzen's system of natural deduction. (Lindahl & Odelstad, 1999a, p. 165.)

Introduction and elimination rules are discussed further in Lindahl & Odelstad (2008a).

The analysis of the concept of 'intermediary' involves complicated questions of meaning and is therefore a philosophically loaded topic. The formal theory of intervenients which is presented in Lindahl & Odelstad (2008a) and (2008b) is intended as a means for a thorough analysis of the concept of an intermediary.

An interesting issue in the discussion of intermediaries is the negation of an intermediate concept. Suppose that a_1 is the ground of the intermediary m and that a_2 is the consequence of m. Let m' be the negation of m, i.e. not-m. Is m' an intermediate concept? If the answer is yes, what can be said about

[11] Introduction and elimination rules are discussed in Lindahl & Odelstad (2008a) with reference to Gentzen.

its grounds and consequences? This question, which is discussed in Lindahl & Odelstad (2008a), is complicated, especially if we turn to open intermediaries.

4.2 Open intermediaries

The concept ‘work of equal value’ is an essential concept in the Swedish Equal Opportunities Act. The following quotation demonstrates this (emphasis added here):

> Employers and employees shall cooperate in pursuing active efforts to promote equality in working life. They shall strive in particular to prevent and eliminate differences in pay and in other conditions of employment between women and men performing *work* that may be considered equal or *of equal value.* They shall also promote equal opportunities for wage growth for women and men.
>
> Work is to be considered *equal in value* to other work if, based on an overall assessment of the nature of the work and the requirements imposed on the worker, it may be deemed to be of similar value. Assessments of work requirements shall take into account criteria such as knowledge and skills, responsibility and effort. When the nature of the work is assessed, particular regard shall be taken of the working conditions.

The concept ‘work of equal value’ is an intermediary with—using the Janus-metaphor—one face looking at the nature of and requirements for the work and the other face looking at efforts to promote equality in working life, especially equal pay for equal work. The law does not supply us with a complete set of introduction rules for the concept. Instead it mentions some criteria that equality of value depends on, viz. knowledge and skills, responsibility and effort. However, one can extract the following uncontroversial introduction rule: if x and y are work that requires the same degree of knowledge, skills, responsibility and effort, then x and y are work of equal value. We can express this in a formalised style as follows:

$$x \sim_1 y \;\&\; x \sim_2 y \;\&\; x \sim_3 y \;\&\; x \sim_4 y \;\&\; x \sim_5 y \;\longrightarrow\; x \sim_v y$$

where
$\sim_1$ is the relation ‘equal knowledge’
$\sim_2$ is the relation ‘equal skills’
$\sim_3$ is the relation ‘equal responsibility’
$\sim_4$ is the relation ‘equal effort’
$\sim_5$ is the relation ‘equal working conditions’
$\sim_v$ is the relation ‘equal value’

Note that the equality relations $\sim_1$,$\sim_2$,$\sim_3$,$\sim_4$ and $\sim_5$ are here regarded as conditions and we can therefore apply Boolean operations on the equality relations, for example construct conjunctions of them. One of the grounds of $\sim_v$ is thus the condition

$$\sim_1 \wedge \sim_2 \wedge \sim_3 \wedge \sim_4 \wedge \sim_5 .$$

But it is also possible that work x and y are of equal value even if they are not equal with respect to the requirements knowledge, skills, responsibility, effort and working condition. We can imagine a situation such that x requires more knowledge than y, and y more responsibility than x but that these two differences balance out. But to turn this observation into an introduction rule is often not possible. The applicability of the concept work of equal value in a certain case must therefore be based on judgments of what holds in the actual case. And even if the law does not state detailed rules for these judgments it gives guidelines, for example in terms of what are possible inputs in such judgments or what factors or circumstances must be taken into account.

The grounds of the concept 'work of equal value' is thus only partially determined by the law in the form of introduction rules. The application of the concept in special cases deserves interpretative decisions based on the role and function of the concept in the law. We call such intermediaries *ground-open*. Concepts such that the consequences are only partially determined by elimination rules are called *consequence-open*.

Open intermediaries are further discussed in Lindahl & Odelstad (2008a). For a detailed discussion of the concept work of equal value, see Odelstad (2008a).

4.3 Intermediaries in normative systems

A normative system is only in rather special cases a two-sorted implicative conceptual system, i.e. a system of grounds and a system of consequences. Instead, normative systems often contain also many intermediate concepts. In more complex normative systems, for example legal systems, there are usually more than one system of intermediaries, and these systems often form a kind of network, where between intermediaries of two different sorts there are intermediaries of a third sort.[12] Note that a rule can simultaneously be an introduction rule for one concept and an elimination rule for another.

Intermediaries do not only exist in normative systems but in many other *msic*-systems. This is discussed in Lindahl & Odelstad (1999a) p. 178.

4.4 A remark on related work

The Scandinavian discussion of intermediate concepts has had a crucial influence on the theory of *msic*-systems put forth in this essay. The following works have been of special significance: Wedberg (1951), Ross (1951), Halldén (1978) and Lindahl (1985). Hedenius (1941) does not consider intermediate concepts but Hedenius' discussion about spurious and genuine norms is of great interest in this context. The works on introduction and elimination rules in logic and philosophy

[12] In Lindahl & Odelstad (2008b), this is illustrated as Figure 1. There the lines between different nodes represent sets of introduction or elimination rules.

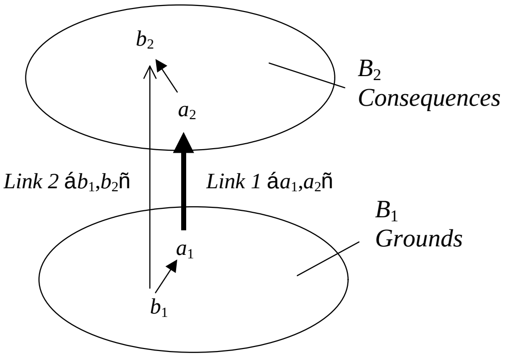


Fig. 3. Norm $\langle a_1,a_2\rangle$ is narrower than norm $\langle b_1,b_2\rangle$.

of mathematics by Gentzen, Dummett and Prawitz have, as emphasized above, also influenced this work. (See Gentzen 1934, Dummett 1973 and Prawitz 1977).

There are similarities between Richard Hare's prescriptivism and the view of intermediaries developed in the work that Lindahl and I have conducted. In Lindahl & Odelstad (1999a), there is a reference to Hare (1989), but the relation between open intermediate concepts and prescriptivism ought to be investigated in more detail.

I have been influenced by P.W. Bridgman's operationalistic approach to concept formation and it seems to me that operationalism and the ideas about intermediate concepts fit well together in roughly the following manner: If a predicative concept is neither purely normative nor operationally definable, consider if it is an intermediate concept. To develop this dictum in detail is not, however, within the scope of the present essay.

5 Implicative closeness between strata

One important problem area in the study of *msic*-systems is the "closeness" between different strata. Some of the ideas regarding this topic will be informally described in this section.

Consider the norms (links) from the system B_1 of grounds to the system B_2 of consequences. One norm can be "narrower" than another, which is illustrated in Figure 3.[13] Suppose that $\langle a_1, a_2\rangle$ and $\langle b_1, b_2\rangle$ are norms from the system of grounds B_1 to the system of consequences B_2.

Figure 3 illustrates that $\langle a_1, a_2\rangle$ is narrower than $\langle b_1, b_2\rangle$. We can say alternatively that $\langle a_1, a_2\rangle$ "lies between" b_1 and b_2. We define the relation 'at least as narrow as', expressed by $\trianglelefteq$, in the following way:

$$\langle a_1, a_2\rangle \trianglelefteq \langle b_1, b_2\rangle \text{ if and only if } b_1\mathcal{R}a_1 \text{ and } a_2\mathcal{R}b_2.$$

[13] See Lindahl & Odelstad (2003) p. 84.

It is easy to see that if $\mathcal{R}$ is a quasi-ordering, i.e. transitive and reflexive, then $\trianglelefteq$ is also a quasi-ordering.

A norm that is *maximally narrow* is *minimal* with respect to the relation 'at least as narrow as'. Hence, a norm $\langle a_1, a_2 \rangle$ is maximally narrow if there is no norm in the system that is strictly narrower than $\langle a_1, a_2 \rangle$, i.e. if $\langle a_1, a_2 \rangle$ is a minimal element with respect to 'at least as narrow as'. In a normative system, the set of norms that are maximally narrow play a crucial role. Given certain requirements of a well-formed normative system, all the other norms of the system are determined by its maximally narrow norms and, therefore, any change of such a system implies a change of at least one maximally narrow norm. This is discussed in Odelstad & Lindahl (2002), Lindahl & Odelstad (2003) and (2008a).

The idea behind intermediaries is that they are intermediate between different strata of concepts and offer narrow links between the strata. It is important to notice that the intermediaries between two strata constitute a stratum itself. The introduction rules of the intermediaries are links from the "bottom stratum" to the "intermediate stratum" and the elimination rules of the intermediaries are links from the "intermediate stratum" to the "top stratum". The introduction rule and the elimination rule of an intermediary constitute narrow links, since the introduction rule determines the weakest ground of the intermediary and the elimination rule the strongest consequence. Intermediate concepts are thus studied in terms of how narrow they are the structure of grounds and the structure of consequences. Generally, the "implicative closeness" between strata is analysed using concepts as minimal joining, weakest ground and strongest consequence. Figure 4 illustrates the two last mentioned notions: a_1 is a weakest ground of m if $b_1 \mathcal{R} m$ implies $a_1 \mathcal{R} b_1$. And a_2 is a strongest consequence of m if $m \mathcal{R} b_2$ implies $a_2 \mathcal{R} b_2$. As a preliminary approximation we can say that the introduction rule of an intermediary states its weakest ground and the elimination rule states its strongest consequence. In Lindahl & Odelstad (2008b), this is discussed in more detail and a rudimentary typology of intermediate concepts is established.

6 Deontic consequences

Let us for a moment return to the simple picture of a normative system consisting of a system of grounds and a system of consequences. The consequences are normative conditions. So far, what we have said about normative conditions is just that they can be constructed by applying a deontic operation to descriptive conditions. There is an extensive literature on deontic operations and it is not intended to enter this discussion here. In this essay, the combination of deontic and action logic developed by Stig Kanger will be used, especially the theory of normative positions created by Kanger and Lindahl.

6.1 Deontic logic with the action operator Do

Kanger exploited the possibilities of combining the deontic operator Shall with the binary action operator Do. The operation Do means that one sees to it

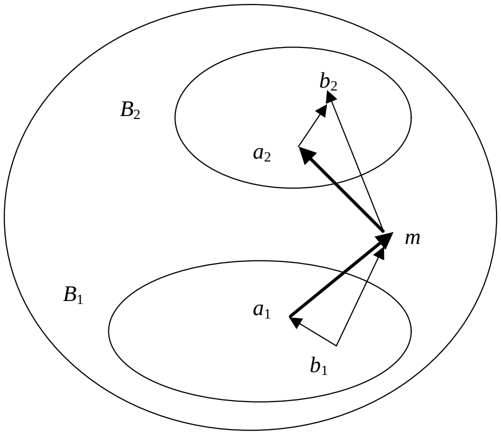


Fig. 4. m is an intermediate concept between B_1 and B_2 with weakest ground a_1 and strongest consequence a_2.

that something is the case (see Kanger, 1957). To be more exact, Shall Do(x, q) means that it shall be that x sees to it that q, while for example $\neg$Shall Do$(y, \neg q)$ means that it is not the case that it shall be that y sees to it that not q. The combination of the deontic operator Shall with the action operator Do and the negation operation $\neg$ gives us a powerful language for expressing purely normative sentences. Kanger emphasized the possibilities of external and internal negation of sentences where these operators are combined. Using combinations of deontic and action operators, we can formulate norms in a more effective way. A conditional norm may for example have the following form: 'If p then it shall be the case that x sees to it that q', which thus can be written as

$$p \rightarrow \text{Shall Do}(x, q).$$

In such norms, p is often a state of affairs which is about x and y, while q is a state of affair which deals with y, i.e. p can be seen as predicate with x and y as variables while q is a predicate with y as the only variable. Hence, a conditional norm can have the following form:

$$p(x, y) \rightarrow \text{Shall Do}(x, \neg q(y)).$$

A concrete example of a norm which has this form is as follows. Suppose that $p(x, y)$ means that x owns y and y is a dog while $q(y)$ means that y fouls public places. The norm above then says that the owner of a dog shall see to it that the dog does not foul in public places.

Note that the sentence May Do(x, q) can be defined in terms of the operators Shall and Do in the following way:

May Do(x, q) if and only if $\neg$Shall $\neg$Do(x, q).

It is worth noting that conditional norms have some similarities with production rules. According to Luger (2002) p. 171, a production rule is

> a *condition-action* pair and defines a single chunk of problem-solving knowledge. The *condition part* of the rule is a pattern that determines when that rule may be applied to a problem instance. The *action part* defines the associated problem-solving step.

The antecedent (or ground) in a norm corresponds to the condition part in a production rule, and the consequent (or consequence) in a norm corresponds to the action part. A production rule thus has the logical form

$p \rightarrow \text{Do } q$

or perhaps better

$p \rightarrow \text{Shall Do } q.$

6.2 Normative positions

In 1913, the American jurist Wesley Newcomb Hohfeld published a work in philosophy of law which has been very influential. It carries the title *Fundamental Legal Conceptions as Applied in Judicial Reasoning* and contains a characterization of eight fundamental legal notions, which were meant to serve as fundamental elements in the analysis of more complex legal relations. Inspired by Hohfeld's work, Kanger developed a theory of normative positions using the deontic-action-language. Kanger's theory of normative positions was originally expressed as a theory of types of rights. He emphasized that the term 'right' has various meanings. For example, if Mrs. x has lent 100 dollars to Mr. y, then x has a right of the simple type Claim against y that she gets back the money she has lent to y. Let

$q_1(x, y) : x$ gets back the money x has lent to y.

The type of right Claim with regard to $q_1(x, y)$ is defined in the following way:

$\text{Claim}(x, y, q_1(x, y)) \quad \text{if and only if} \quad \text{Shall Do}(y, q_1(x, y)).$

This means that y shall see to it that x gets back the money she lent y. Further, Mrs. x has probably a right of type Immunity to walk outside Mr. y's shop. Let

$q_2(x, y) : x$ walks outside y's shop

Immunity with regard to q_2 is defined as follows:

$\text{Immunity}(x, y, q_2(x, y)) \quad \text{if and only if} \quad \text{Shall } \neg\text{Do}(y, \neg q_2(x, y)).$

Hence, it shall be the case that y does not see to it that x does not walk outside y's shop. (These examples are taken from Lindahl, 1994, p. 891-892.)

Kanger's work was considerably improved and extended into a formal theory of normative positions in Lindahl (1977). Lindahl developed three systems of types of normative positions. The simplest one is the system of one-agent types of normative position, and only this system is used in this essay. The one-agent types are constructed in the following way. Let $\pm\alpha$ stand for either of α or $\neg\alpha$. Starting from the scheme $\pm\text{May}\pm\text{Do}(x, \pm q)$, where $\pm$ stands for the two alternatives of affirmation or negation, a list is made of all maximal and consistent conjunctions, 'maxiconjunctions', such that each conjunct satisfies the scheme.[14] Maximality means that if we add any further conjunct, satisfying the scheme, then this new conjunct either is inconsistent with the original conjunction or redundant. Note that the expression $\neg\text{Do}(x,q)\& \neg\text{Do}(x,\neg q)$ expresses x's passivity with regard to q. Here this expression is abbreviated as $\text{Pass}(x,q)$. By this procedure, the following list of seven maxiconjunctions is obtained, which are denoted $\mathbf{T}_1(x,q)$,...,$\mathbf{T}_7(x,q)$, see Lindahl (1977), p. 92.

$\mathbf{T}_1(x,q)$: $\text{MayDo}(x,q)$ & $\text{MayPass}(x,q)$ & $\text{MayDo}(x,\neg q)$.
$\mathbf{T}_2(x,q)$: $\text{MayDo}(x,q)$ & $\text{MayPass}(x,q)$ & $\neg\text{MayDo}(x,\neg q)$.
$\mathbf{T}_3(x,q)$: $\text{MayDo}(x,q)$ & $\neg\text{MayPass}(x,q)$ & $\text{MayDo}(x,\neg q)$.
$\mathbf{T}_4(x,q)$: $\neg\text{MayDo}(x,q)$ & $\text{MayPass}(x,q)$ & $\text{MayDo}(x,\neg q)$.
$\mathbf{T}_5(x,q)$: $\text{MayDo}(x,q)$ & $\neg\text{MayPass}(x,q)$ & $\neg\text{MayDo}(x,\neg q)$.
$\mathbf{T}_6(x,q)$: $\neg\text{MayDo}(x,q)$ & $\text{MayPass}(x,q)$ & $\neg\text{MayDo}(x,\neg q)$.
$\mathbf{T}_7(x,q)$: $\neg\text{MayDo}(x,q)$ & $\neg\text{MayPass}(x,q)$ & $\text{MayDo}(x,\neg q)$.

$\mathbf{T}_1$,...,$\mathbf{T}_7$ are called the types of one-agent positions.[15] Given the underlying logic, the one-agent types are mutually disjoint and their union is exhaustive. i.e. constitute a partition. Note that $\neg\text{MayDo}$ (x,q) & $\neg$ MayPass (x,q) & $\neg\text{MayDo}(x,\neg q)$ is logically false, according to the logic of Shall and May.

It is easy to see that the last three types can more concisely be described as follows:

$\mathbf{T}_5(x,q)$: Shall $\text{Do}(x,q)$.
$\mathbf{T}_6(x,q)$: Shall $\text{Pass}(x,q)$.
$\mathbf{T}_7(x,q)$: Shall $\text{Do}(x,\neg q)$.
Note that the following "symmetry principles" hold (Lindahl, 1977, p. 92):
$\mathbf{T}_1(x,q)$ if and only if $\mathbf{T}_1(x,\neg q)$
$\mathbf{T}_2(x,q)$ if and only if $\mathbf{T}_4(x,\neg q)$
$\mathbf{T}_3(x,q)$ if and only if $\mathbf{T}_3(x,\neg q)$
$\mathbf{T}_5(x,q)$ if and only if $\mathbf{T}_7(x,\neg q)$
$\mathbf{T}_6(x,q)$ if and only if $\mathbf{T}_6(x,\neg q)$

In Lindahl & Odelstad (2004) and Odelstad & Boman (2004) the one-agent-types in the Kanger-Lindahl theory of normative positions are used as operators on descriptive conditions to get deontic conditions. As a simple example, suppose that r is a unary condition. Then T_ir (with $1 \leq i \leq 7$) is the binary condition such that

$$T_ir(y,x) \text{ iff } \mathbf{T}_i(x,r(y)),$$

where $\mathbf{T}_i(x,r(y))$ is the ith formula of one-agent normative positions. Note that for example $\mathbf{T}_3(x,r(y))$ means

[14] The notion of 'maxiconjunction' was introduced in Makinson (1986), p. 405f.

[15] Formally, a "type" $\mathbf{T}_i$ (1≤i≤7) of one-agent positions refers to the set of all ordered pairs $\langle x,q\rangle$ such that $\mathbf{T}_i(x,q)$.

$$\text{MayDo}(x, r(y)) \ \& \ \neg\text{MayPass}(x, r(y)) \ \& \ \text{MayDo}(x, \neg r(y)).$$

T_i is called a one-agent position-operator. If $\langle p, T_i r\rangle$ is a norm, then from $p(x_1, x_2)$ we can, by using the norm, infer $T_i r(x_1, x_2)$ and thus also $\mathbf{T}_i(x_2, r(x_1))$, which means that, with regard to the state of affairs $r(x_1)$, x_2 has a normative position of type $\mathbf{T}_i$.

The theory of normative positions was developed during the 60s and 70s, primarily as an analytical tool to be used in jurisprudence and political science. The Kanger-Lindahl theory of normative positions was applied to problems in computer science in the 90s, see Jones & Sergot (1993) and (1996), Sergot (1999) and (2001), Krogh (1995) and Krogh & Herrestad (1999).

6.3 Normative systems as *msic*-systems

Conceiving of normative systems as *msic*-systems is a kind of representation of normative systems. What characterizes the subclass of normative systems among *msic*-systems in general are their cognitive features. A normative system consists of one stratum of descriptive grounds and another stratum of normative consequences and eventually one or more strata of intermediaries. Furthermore, a normative system contains links or joinings between the strata. Note that the final consequences are expressed in terms of normative conditions, for example constructed by applying deontic operations to descriptive conditions. Thus, representing normative systems in this way puts the emphasis on concepts and not on propositions.

7 The algebraic approach to *msic*-systems

The study of the structure of *msic*-systems, especially the implicative closeness between different strata, is one of the main goals of a series of papers co-authored together with Lars Lindahl (see References for details).[16] As tools for this endeavour, algebraic concepts and theories are used. In this section, two of the structures that play a crucial role as such tools will be described briefly. But first a preliminary remark.

7.1 Set-theoretical predicates

A common way of characterizing formal theories in mathematics is described by Suppes as follows:

[16] See especially Lindahl & Odelstad (2003), (2004), (2008a) and (2008b). Technical results in our papers include a characterization of an *msic*-system in terms of the most narrow joinings between different strata, characterization of the structure of the most narrow joinings between two strata, conditions for the extendability of intermediate concepts, and finally, a specification of the conditions such that the Boolean operations on intermediate concepts will result in intermediate concepts and characterization of most narrow joinings in terms of weakest grounds and strongest consequences.

> The kernel of the procedure for axiomatizing theories within set theory may be described very briefly: to axiomatize a theory is to define a predicate in terms of notions of set theory. A predicate so defined is called a *set-theoretical* predicate. (Suppes, 1957, p. 249.)

A simple example of a set-theoretical predicate is 'to be a quasi-ordering':

Definition 1. *Let A be a set and R a binary relation on A. The relational structure $\langle A, R\rangle$ is a* quasi-ordering *if for all a, b, c in A, the following axioms are satisfied:*
(1) aRa (reflexivity)
(2) If aRb and bRc, then aRc (transitivity).

'To be a quasi-ordering' is a predicate, which is true or false of relational structures. This set-theoretical predicate characterizes an axiomatized theory, the theory of quasi-orderings, and a model of that theory is a structure satisfying the predicate 'to be a quasi-ordering'.

Two set-theoretical predicates which play a crucial role in the *msic*-theory will now be presented.

7.2 Boolean quasi-orderings and joining systems

Definition 2. *The relational structure $\langle B, \wedge, ', R\rangle$ is a* Boolean quasi-ordering (Bqo) *if $\langle B, \wedge, '\rangle$ is a Boolean algebra, R is a quasi-ordering, $\bot$ is the zero element, $\top$ is the unit element and R satisfies the additional requirements:*
(1) aRb and aRc implies $aR(b \wedge c)$,
(2) aRb implies $b'Ra'$,
(3) $(a \wedge b)Ra$,
(4) not $\top R\bot$.

Boolean algebras are well-known structures with many applications. A Boolean quasi-ordering is a quasi-ordering defined on a Boolean algebra in such a way that it determines a new Boolean algebra related to the first one in a special way. This is explained in more detail in Lindahl & Odelstad (2004). The definition of a Boolean joining system, which follows below, presupposes the definition of a Boolean quasi-ordering. Many normative systems can be represented as Boolean joining systems or combinations of two or more such systems. First a reminder of a notion discussed earlier:

Definition 3. *The* narrowness-relation determined by *the quasi-orderings $\langle B_1, R_1\rangle$ and $\langle B_2, R_2\rangle$ is the binary relation $\trianglelefteq$ on $B_1 \times B_2$ such that $\langle a_1, a_2\rangle \trianglelefteq \langle b_1, b_2\rangle$ if and only if $b_1 R_1 a_1$ and $a_2 R_2 b_2$.*

Note that $\trianglelefteq$ is a quasi-ordering on $B_1 \times B_2$.

Definition 4. *A* Boolean joining system (Bjs) *is an ordered triple $\langle \mathcal{B}_1, \mathcal{B}_2, J\rangle$ such that $\mathcal{B}_1 = \langle B_1, \wedge, ', R_1\rangle$ and $\mathcal{B}_2 = \langle B_2, \wedge, ', R_2\rangle$ are Bqo's and $J \subseteq B_1 \times B_2$, and the following requirements are satisfied:*

(1) for all $b_1, c_1 \in B_1$ *and* $b_2, c_2 \in B_2$, $\langle b_1, b_2 \rangle \in J$ *and* $\langle b_1, b_2 \rangle \trianglelefteq \langle c_1, c_2 \rangle$ *implies* $\langle c_1, c_2 \rangle \in J$,
(2) for any $C_1 \subseteq B_1$ *and* $b_2 \in B_2$, *if* $\langle c_1, b_2 \rangle \in J$ *for all* $c_1 \in C_1$, *then* $\langle a_1, b_2 \rangle \in J$ *for all* $a_1 \in lub_{R_1} C_1$,
(3) for any $C_2 \subseteq B_2$ *and* $b_1 \in B_1$, *if* $\langle b_1, c_2 \rangle \in J$ *for all* $c_2 \in C_2$, *then* $\langle b_1, a_2 \rangle \in J$ *for all* $a_2 \in glb_{R_2} C_2$.

A norm can, as has been pointed out above, in many contexts be regarded as consisting of two objects, a ground condition and a consequence condition standing in an implicative relation to each other. The ground belongs to one Boolean quasi-ordering and the consequence to another. Therefore, we can view a normative system as a set of joinings of a Boolean quasi-ordering of grounds to a Boolean quasi-ordering of consequences, where $\wedge$ and $'$ are Boolean operations on the conditions. A normative system $\mathcal{N}$ can therefore be represented as a Boolean joining system $\langle \mathcal{B}_1, \mathcal{B}_2, J \rangle$ where $\mathcal{B}_1 = \langle B_1, \wedge, ', R_1 \rangle$ is a Boolean quasi-ordering of ground-conditions, $\mathcal{B}_2 = \langle B_2, \wedge, ', R_2 \rangle$ a Boolean quasi-ordering of consequence-conditions and the set J, where $J \subseteq B_1 \times B_2$, is the set of norms. Note that the implicative relation in the system $\mathcal{N}$ is represented in the different parts of the system by the relations R_1, R_2 and J respectively.

It is worth noting that there is a difference in notational conventions between the definition of a *Bqo* and the definition of a *Bjs*. In a *Bqo*, if the relation R holds between a and b this is written aRb. If in a *Bjs* J holds between a_1 and a_2 this is written $\langle a_1, a_2 \rangle \in J$. The reason is that in the intended models of *Bjs*'s, the elements in J are treated as objects in a way that does not hold for the elements in R. In a representation of a *Bjs* as a normative system, $\langle a_1, a_2 \rangle \in J$ means that the norm $\langle a_1, a_2 \rangle$ holds in the system, and the elements in J are subject to comparison with respect to, for example, narrowness.

Given the narrowness relation $\trianglelefteq$ one can determine the set of minimal elements of J, $\min J$, with respect to $\trianglelefteq$. Under fairly general conditions, the set $\min J$ characterizes J in the following way:

$$\langle a_1, a_2 \rangle \in J \ \text{ iff } \ \exists \langle b_1, b_2 \rangle \in \min J : \langle b_1, b_2 \rangle \trianglelefteq \langle a_1, a_2 \rangle .$$

Given certain general presuppositions, one can choose a subset C of $\min J$ from which $\min J$ can be inferred and which therefore also determines J. We call such a set C a *base of minimal elements* of J. In many contexts, the elements in C can be represented by intermediate concepts. An intermediary is determined by the condition that constitute its maximally narrow ground and the condition that constitutes its maximally narrow consequence. See Lindahl & Odelstad (2008a) and (2008b) for further details.

7.3 Models and variations of the algebraic theories

As has been emphasized in earlier sections with normative systems as a key example, one approach to the representation of *msic*-systems is by regarding concepts as conditions subject to Boolean operations and with an implicative relation defined on these conditions. A *Bqo* or a *Bjs* with domains of conditions is called a *condition implication structure*, abbreviated *cis.* A special kind of *cis*-representation of a normative system is the *npcis*-representation of normative conditions. In an *npcis,* a normative condition is constructed by applying the one-agent position-operators to descriptive conditions (see Lindahl & Odelstad, 2004).

There are some limitations of the *cis*-representation of *msic.* One problem is the formation of conjunctions and disjunctions of conditions of different arity. How this can be handled is discussed in Lindahl & Odelstad (2004) section 3. Another weakness of the *cis*-representation is that new conditions can only be constructed out of given conditions by Boolean operations. As a consequence, it is, for example, not possible to define within a *cis* the condition 'to be the grandfather of' in terms of the conditions 'to be the father of' and 'to be the mother of'. Note that if we want 'grandfather' to be a condition in our *cis* we can of course include it as a primitive condition.

With reference to the limitations mentioned above, it might be held that the *cis*-representation is too simple to be suitable for an overall representation of an actual legal system or a complex *msic*-systems of some other kind. Nevertheless, the *cis*-representation is sufficiently rich to permit a detailed study of a number of issues pertaining especially to intermediate concepts in a legal system.[17] The *cis*-representation can in a sense be viewed as an "idealized model" for studying different phenomena in *msic*-systems. When judging the usefulness of the *cis*-representation it is worth noting the following: Even if there are a number of difficulties when it comes to a detailed representation of norms as joinings in a Boolean joining system, it may be the case that these difficulties do not appear when the objective in view is rather to construct an artificial normative system regulating an artificial multiagent-system.

Condition implication structures are not the only kind of models of Boolean joining systems that are interesting as representations of *msic*-systems. It is easy to see that we can construct a *Bqo* out of a first order theory Σ. Consider the structure $\langle B, \wedge, ', R\rangle$ where $\langle B, \wedge, '\rangle$ is the Lindenbaum algebra of the predicate calculus. Let R be the quasi-ordering on B determined by the Lindenbaum algebra of Σ. Then $\langle B, \wedge, ', R\rangle$ is a Boolean quasi-ordering.

Boolean joining systems are obviously based on the notion of a Boolean algebra. However, it is possible to define an analogous kind of systems based on lattices. Such a system $\langle \mathcal{L}_1, \mathcal{L}_2, J\rangle$ consists of the latticed quasi-orderings $\mathcal{L}_1 = \langle L_1, \wedge, \vee, R_1\rangle$ and $\mathcal{L}_2 = \langle L_2, \wedge, \vee, R_2\rangle$ and the set J of joinings between them and can be called a *latticed joining system,* abbreviated *Ljs.* A large fraction of the formal result proved for *Bjs*'s will hold also for *Ljs*'s, roughly because the complement operation in the Boolean algebras does not play a role in the proofs. There are models of the theory of *Ljs* that can be interesting representations of *msic*-systems. This holds, for example, when the concepts in the *msic*-systems

[17] Cf. Lindahl & Odelstad (2006b), where it is suggested that a representation based on cylindric algebras would be more appropriate than a representation based on Boolean algebras.

are not conditions but instead for instance aspects or equality relations for aspects.

7.4 The formal representation of *msic*-systems

The formal theory of *msic*-systems is to a large extent a question of representation. The algebraic framework for the representation of *msic*-systems in the work that Lindahl and I have conducted has gone through different "stages" and I will outline and discuss these stages here.

Stage 1: Lattice-representation In Lindahl & Odelstad (1996) and (1999a), an *msic*-system is represented as a lattice $\langle L, \leq \rangle$ of conditions extended with a quasi-ordering ρ. The lattice operations represent conjunction and disjunction respectively. Negation is not included for purely pragmatic reasons; in the first version of the theory we preferred to simplify the matter but still be able to express our main ideas about intermediaries. The partial ordering $\leq$ in the lattice represents "logical implication" and the quasi-ordering ρ represent implications in a more general sense. The relation between the partial ordering $\leq$ and the quasi-ordering ρ is such that the partial ordering $\leq_\rho$ generated from ρ by the formation of equivalence classes is a lattice and $\leq$ is a subset of $\leq_\rho$. $\langle L_\rho, \leq_\rho \rangle$ is the quotient algebra of $\langle L_\rho, \leq_\rho \rangle$ with the respect to the indifference part of ρ. A two-sorted conceptual system is represented as a system of two sublattices $\langle L_1, \leq_1 \rangle$ and $\langle L_2, \leq_2 \rangle$ of $\langle L, \leq \rangle$ and the set $\{\langle x_1, x_2 \rangle \in L_1 \times L_2 \mid x_1 \leq_\rho x_2\}$ of joinings between the sublattices.

Stage 2: *Bqo*-representation In Odelstad & Lindahl (1998), the formal framework for representing *msic*-systems is modified in some respects:
(1) We incorporate the operation of negation and suppose that the conditions constitute a Boolean algebra $\langle B, \wedge, ' \rangle$.
(2) We do not make a transition to the quotient algebra of $\langle B, \wedge, ' \rangle$ with respect to the indifference part of ρ. Instead we construct the Boolean quasi-ordering $\langle B, \wedge, ', \rho \rangle$. The reason is that we want to distinguish between two conditions even if they are indifferent with respect to ρ. See Lindahl & Odelstad (2004) section 2.1.
(3) We make a clearer separation between the algebraic theories and the models used for the representation of *msic*-systems. In stage 1, we regarded the lattice operations $\wedge$ and $\vee$ as representing conjunction and disjunction of conditions, since we only had one intended model in view. A *Bqo* of conditions is one kind of *Bqo*-model which can be used for representing *msic*-systems and we do not exclude the possibility that there can be other kinds of models.

Note that an *msic-system* is represented as a system of substructures of $\langle B, \wedge, ', \rho \rangle$, called fragments, and the set of joinings between them. The formal tools for the representation of *msic*-systems based on the *Bqo*-theory is further developed in Odelstad & Lindahl (2000), Lindahl and Odelstad (2000) and (2004). This *Bqo*-representation is used in Odelstad & Boman (2004) and Lindahl & Odelstad (2003).

Stage 3: *Bjs*-representation In the *Bqo*-representation of *msic*-systems, strata of concepts of different sorts are represented as fragments of the basic *Bqo* $\langle B, \wedge, ', \rho\rangle$. Hence, B contains conditions of different sorts. But B contains also Boolean combinations of concepts of different sorts, i.e. compound concepts of a "mixed sort". In many contexts, however, concepts of such mixed sorts are not of any interest and make the situation unnecessarily complicated. To avoid this complication, a *Bjs* can be a useful tool for representations. A two-sorted conceptual system is then represented as a *Bjs* $\langle \mathcal{B}_1, \mathcal{B}_2, J\rangle$ consisting of two *Bqo*'s $\mathcal{B}_1$ and $\mathcal{B}_2$ together with the set of joinings J between them. The *Bqo*'s $\mathcal{B}_1$ and $\mathcal{B}_2$ are not necessarily fragments of one *Bqo*. The axioms of a *Bjs* are such that two fragments of a *Bqo* and the joinings between them constitute a *Bjs*.

However, if one wants to study *msic*-systems containing conditions of several different sorts, this would involve a number of *Bjs*'s related to each other in a complicated way. It may then be useful to have as a background a Boolean algebra $\langle B, \wedge, '\rangle$ representing the "language" of the *msic*-system and a binary relation ρ representing the non-logical (for example normative) content of the system. The sets of joinings between different strata of concepts will then be contained in ρ. A *msic*-system may therefore appropriately be represented as a *supplemented Boolean algebra*, abbreviated *sBa*, $\langle B, \wedge, ', \rho\rangle$ with *Bjs*'s lying within it. This is the approach in Lindahl & Odelstad (2008a) and (2008b).

7.5 Non-Boolean joining systems

In this section, two examples of joining systems consisting of concepts but not constituting *Bjs* will be outlined briefly.

Joining systems of equality-relations In this essay I have focused on *msic*-systems where the concepts are conditions subject to the Boolean operations. But there are kinds of conditions that do not constitute Boolean algebras. One example is equality-relations. The term 'equality-relation' here refer to a relation of equality with respect to some aspect α, and it is presupposed in this context that an equality-relation is always an equivalence-relation, i.e. a reflexive, transitive and symmetric relation. Let A be a non-empty set and let $E(A)$ be the set of equivalence relations on A. Define the binary relation $\leq$ on $E(A)$ in the following way: For all $\varepsilon_1, \varepsilon_2 \in E(A)$

$$\varepsilon_1 \leq \varepsilon_2 \text{ iff } x\varepsilon_1 y \text{ implies } x\varepsilon_2 y. \tag{1}$$

The reader should be reminded of the following well-known fact. $\mathcal{E}(A) = \langle E(A), \leq\rangle$ is a complete lattice. Note that the negation ε' of an equivalence relation $\varepsilon \in E(A)$ is not an equivalence relation, i.e. $\varepsilon' \notin E(A)$. Let $\mathcal{E}_1 = \langle E_1, \leq_1\rangle$ and $\mathcal{E}_2 = \langle E_2, \leq_2\rangle$ be disjoint complete sublattices of $\mathcal{E}(A)$ and consider $\langle \mathcal{E}_1, \mathcal{E}_2, J\rangle$ where $J = \leq / (E_1(A) \times E_2(A))$. Given some general conditions $\langle \mathcal{E}_1, \mathcal{E}_2, J\rangle$ is a joining system. We have here an example of a joining system which consists of conditions but they do not constitute a Boolean algebra.

A Boolean quasi-ordering is a Boolean algebra extended with a quasi-ordering satisfying certain conditions. We can define an analogous structure based on a lattice instead of a Boolean algebra. Let $E(A)$ and $\leq$ be as above and let $\langle E(A), \wedge, \vee\rangle$ be the lattice $\langle E(A), \leq\rangle$ expressed in terms of operations instead of a partial ordering, i.e. $\varepsilon_1 \wedge \varepsilon_2 = \inf\{\varepsilon_1, \varepsilon_2\}$ and $\varepsilon_1 \vee \varepsilon_2 = \sup\{\varepsilon_1, \varepsilon_2\}$. Suppose that R is a quasi-ordering on $E(A)$ such that

(1) aRb and aRc implies $aR(b \wedge c)$.
(2) aRc and bRc implies $(a \vee b)Rc$.
(3) $(a \wedge b)Ra$.
(4) $aR(a \vee b)$.

Then $\langle E(A), \wedge, \vee, R\rangle$ is called a *latticed quasi-ordering.* The transition to the quotient algebra of $\langle E(A), \wedge, \vee\rangle$ with respect to the indifference part of R will result in a lattice. (Cf. Lindahl & Odelstad, 1999a, p. 171.) The *msic*-systems consisting of equality-relations can often be represented as latticed quasi-orderings, and this also holds for *msic*-systems consisting of aspects.

Joining systems of aspects As pointed out above, this essay has focused on *msic*-systems where the concepts are conditions. But there are other kinds of concepts, for example aspects, in many disciplines called attributes. As examples of aspects let me mention a few: area, temperature, age, loudness and archeological value. It is a common view of aspects that they can, in some way or another, be represented as relational structures. In Odelstad (1992), a theory of aspects, where aspects are represented by systems of relationals, is set out. A relational is a function with sets as arguments and structures as values. On sets of systems of relationals, several quasi-orderings can be defined but here only one example will be given.

Let Rels $\mathcal{D}$ denote the set of systems of relationals whose range of definition is the family $\mathcal{D}$ of sets. This means that for all $\Re \in$ Rels $\mathcal{D}$ it holds that $\Re = \langle\rho_i\rangle_{i\in I}$ for some set I and for all $A \in \mathcal{D}$, $\rho_i(A) \subseteq A^{\nu_i}$ where ν_i is the arity of the relational ρ_i. Hence, $\Re(A) = \langle A, \rho_i\rangle_{i\in I}$. Let $\Im(\Re(A), \Re(B))$ denote the set of isomorphisms from $\Re(A)$ to $\Re(B)$. We can define a relation *sub* on Rels $\mathcal{D}$ in the following way: If $\Re_1, \Re_2 \in$ Rels $\mathcal{D}$ then

$$\Re_2 \; sub \; \Re_1 \;\text{ iff }\; \text{for all } A, B \in \mathcal{D} : \Im(\Re_2(A), \Re_2(B)) \supseteq \Im(\Re_1(A), \Re_1(B)). \quad (2)$$

It is obvious that *sub* is a quasi-ordering on Rels $\mathcal{D}$. It follows from Odelstad (1992) that $\langle \text{Rels}\,\mathcal{D}, sub\rangle$ is a complete quasi-lattice and it is therefore possible that there are joining systems lying within $\langle \text{Rels}\,\mathcal{D}, sub\rangle$.[18] The relational

[18] If $\langle A, R\rangle$ is a quasi-ordering such that $\text{lub}_R\{a, b\} \neq \varnothing$ and $\text{glb}_R\{a, b\} \neq \varnothing$ for all $a, b \in A$, then $\langle A, R\rangle$ will be called a *quasi-lattice.* If $\text{lub}_R X \neq \varnothing$ and $\text{glb}_R X \neq \varnothing$ for all $X \subseteq A$, then a quasi-ordering $\langle A, R\rangle$ is a *complete quasi-lattice.*

Suppose that $\langle A, R\rangle$ is a quasi-lattice, Q the equality-part of R and A_Q is the set of Q-equivalence classes generated by elements of A. Then $\langle A_Q, \rho\rangle$, where $[a]_Q\, \rho\, [b]_Q$ iff aRb, is a lattice. If $\langle A, R\rangle$ is a complete quasi-lattice then $\langle A_Q, \rho\rangle$ is a complete lattice.

systems in $\langle \mathrm{Rels}\,\mathcal{D}, sub\rangle$ can be of different sorts and it is a meaningful question if they form joining systems or even latticed joining systems. Note that in $\langle \mathrm{Rels}\,\mathcal{D}, sub\rangle$ the implicative relation *sub* is not implication in the usual sense but expresses a kind of dependence relation.

7.6 A remark on input-output logic

In a series of papers, Makinson and van der Torre have developed a highly interesting theory called input-output logic, see for example Makinson and van der Torre (2000) and (2003). One striking similarity between input-output logic and the theory of *msic*-systems is that norms are represented as ordered pairs. This observation raises the question if there are some deep similarities between input-output logic and *msic*-theory. However, let me first state some obvious differences between the two theories. While *msic*-systems are by definition at least two-sorted, this does not holds for input-output logic. A common feature of the study of *msic*-systems reported here is the implicative closeness between strata of different sorts in an *msic*-system. An analogous study does not seem to have been carried out for input-output logic. The strata of an *msic*-system of conditions are Boolean structures (*Bqo's* to be more precise), but the strata of *msic*-systems of other kinds need not be Boolean structures; instead, they can for example be lattice-like structures. In input-output logic, the set of inputs constitute a Boolean algebra and the same holds for the set of outputs.

The following remark sheds some light on the relation between input-output logic and the theory of *msic*-systems. (Knowledge of input-output logic is presupposed.) Suppose that $\langle \mathcal{B}_1, \mathcal{B}_2, J\rangle$ is a *Bjs* where $\mathcal{B}_1 = \langle B_1, \wedge, ', R_1\rangle$ and $\mathcal{B}_2 = \langle B_2, \wedge, ', R_2\rangle$. Makinson and van der Torre state a number of rules for the output operators they define. Translated to a *Bjs* these rules are as follows:

Strengthening Input: From $\langle a_1, a_2\rangle \in J$ to $\langle b_1, a_2\rangle \in J$ whenever $b_1 R_1 a_2$.
Follows from condition (1) of a *Bjs*.
Conjoining Input: From $\langle a_1, a_2\rangle \in J$ and $\langle a_1, b_2\rangle \in J$ to $\langle a_1, a_2 \wedge b_2\rangle \in J$.
Follows from condition (3) of a *Bjs*.
Weakening Output: From $\langle a_1, a_2\rangle \in J$ to $\langle a_1, b_2\rangle \in J$ whenever $a_2 R_2 b_2$.
Follows from condition (1) of a *Bjs*.
Disjoining Input: From $\langle a_1, a_2\rangle \in J$ and $\langle b_1, a_2\rangle \in J$ to $\langle a_1 \vee b_1, a_2\rangle \in J$.
Follows from condition (2) of a *Bjs*.

There are three conditions on a joining space in a Boolean joining system. The comparison with input-output logic above shows that it could be of interest to define weaker kinds of systems characterized by, for example, condition (1) and (3).

8 Applications of *msic*-systems in agent theory

8.1 Introduction

The applications of the theory of *msic*-systems in computer science can follow different paths. One path goes through the representation of normative systems as *msic*-systems and the applications of normative systems in computer science. Along a related path, the focus is on intermediate concepts, which are important in normative systems but also in other kinds of systems, for example in knowledge representation systems. A third path is the use of conceptual structures in fields like the Semantic Web and information extraction. Here a few comments on the use of *msic*-systems in the theory of artificial agents will be made, where the *msic*-systems will mainly represent normative systems.

8.2 Agent *oeconomicus norma*

Within economic theory the consumer's behaviour has traditionally been described as determined by a utility function. During the last three decades there has been a growing interest among researchers in how norms (for example rules of law) pose restrictions on the behaviour induced by the utility function. The behaviour of the consumers or other economic agents, according to this model, is the result of the interplay between optimization of the utility function and restrictions due to norms. We may perhaps speak of *norm-regulated Homo oeconomicus.* It has also been suggested that a model of this kind could be used for regulating the behaviour of artificial agents. We can perhaps call this model *Agent oeconomicus norma.* The role that norms will have in regulating the behavior of agents is, according to this model, to delimit the autonomy of the agents. Metaphorically one can say that the norms define the scope (*Spielraum*) for an agent. The agent chooses the act it likes best within the scope determined by the norms.

Norm-regulation of agents presupposes a precise and significant representation of norms and normative systems. As was explained in previous sections, a norm is here represented as an implicative sentence where the antecedent is a descriptive condition stating the circumstances of an agent, and the consequent is a condition expressing the normative or deontic position that the agent has with respect to a state of affairs. Hence, from the norms of the system will follow a deontic structure over possible state of affairs implying that some states may be permissible while the rest are non-permissible. The "wish" or "desire" of an agent is represented as a preference structure over possible states or situations. The agent chooses an act which leads to one of the permissible states that it prefers the most.

In Odelstad & Boman (2004), the ideas outlined above were developed using the typology of normative (deontic) positions developed by Kanger and Lindahl and the algebraic representation of normative systems that Lindahl and I have developed. The aim of Odelstad & Boman (2004) was to present a model of how norms can be used to regulate the behaviour of multiagent-systems on the assumption that the role of norms is to define the *Spielraum* for an agent.[19] An abstract architecture was defined in terms of a set-theoretical predicates and a Mas (a multiagent-system) having this architecture is called a norm-regulated Dalmas.[20] One of the results in Odelstad & Boman (2004) was a scheme for how normative positions will restrict the set of actions that the agents are permitted to choose from.

8.3 Normative positions regulating actions

A Dalmas is an ordered 7-tuple $\langle \Omega, S, A, \mathcal{A}, \Delta, \Pi, \Gamma \rangle$ containing

- an agent set Ω ($\omega, \varkappa, \omega_1, ...$ elements in Ω),
- a state or phase space S ($r, s, s_1, ...$ elements in S),
- an action set A such that for all $a \in A$, $a : \Omega \times S \to S$ such that $a(\omega, r) = s$ means that if the agent ω performs the act a in state r, then the result will be state s ($a, b, a_1, ...$ elements in A),
- a function $\mathcal{A} : \Omega \times S \to \wp(A)$ where $\wp(A)$ is the power set of A; $\mathcal{A}(\omega, s)$ is the set of acts accessible (feasible) for agent ω in state s,
- a deontic structure-operator $\Delta : \Omega \times S \to \mathcal{D}$ where $\mathcal{D}$ is a set of deontic structures of the same type with subsets of A as domains and $\Delta(\omega, s)$ is ω's deontic structure on $\mathcal{A}(\omega, s)$ in state s,
- a preference structure-operator $\Pi : \Omega \times S \to \mathcal{P}$ where $\mathcal{P}$ is a set of preference structures of the same type with subsets of A as domains and $\Pi(\omega, s)$ is ω's preference structure on $\mathcal{A}(\omega, s)$ in state s,
- a choice-set function $\Gamma : \Omega \times S \to \wp(A)$ where $\Gamma(\omega, s)$ is the set of actions for ω to choose from in state s.

Note that in the definition the Cartesian product $\Omega \times S$ motivates the introduction of a name for the elements in $\Omega \times S$. Let $\mathfrak{D}$ be a Dalmas. A *situation* for the system $\mathfrak{D}$ is determined by the agent to move ω and the state s. A situation is represented by an ordered pair $\langle \omega, s \rangle$. The set of situations for $\mathfrak{D}$ is thus $\Omega \times S$.

The idea behind a norm-regulated Dalmas is roughly the following: What is permissible for an agent to do in a situation $\langle \omega, s \rangle$ is determined by a normative system $\mathcal{N}$. This idea can be explicated in the following way. Let

$$T_i d(\omega_1, ..., \omega_\nu, \omega; \omega, s) \tag{3}$$

mean that in the situation where it is the agent ω's turn to draw and the state of the system is s, ω has the normative position of type $\mathbf{T}_i$ with regard to the state of affairs $d(\omega_1, ..., \omega_\nu)$.

Prohibited$_{\omega,s}(a)$ means that in the situation where it is ω's turn to draw and the state of the system is s, ω is prohibited to execute the act a.

[19] For the use of the term 'Spielraum' in this context, see Lindahl (1977) and Lindahl (2005).

[20] The term Dalmas is chosen since the architecture is constructed for the application of *d*eontic-*a*ction *l*ogic.

The following seven principles establish connections between the condition $T_i d$ and the predicate Prohibited (see Odelstad & Boman, 2004, p. 160f.):

1. From $T_1 d(\omega_1, ..., \omega_\nu, \omega; \omega, s)$ follows no restriction on the acts.
2. From $T_2 d(\omega_1, ..., \omega_\nu, \omega; \omega, s)$ follows that
   if $d(\omega_1, ..., \omega_\nu; s)$ and $\neg d(\omega_1, ..., \omega_\nu; a(\omega, s))$ then $\text{Prohibited}_{\omega,s}(a)$.
3. From $T_3 d(\omega_1, ..., \omega_\nu \omega; \omega, s)$ follows that
   if $[d(\omega_1, ..., \omega_\nu; s)$ iff $d(\omega_1, ..., \omega_\nu; a(\omega, s))]$ then $\text{Prohibited}_{\omega,s}(a)$.
4. From $T_4 d(\omega_1, ..., \omega_\nu, \omega; \omega, s)$ follows that
   if $\neg d(\omega_1, ..., \omega_\nu; s)$ and $d(\omega_1, ..., \omega_\nu; a(\omega, s))$ then $\text{Prohibited}_{\omega,s}(a)$.
5. From $T_5 d(\omega_1, ..., \omega_\nu, \omega; \omega, s)$ follows that
   if $\neg d(\omega_1, ..., \omega_\nu; a(\omega, s))$ then $\text{Prohibited}_{\omega,s}(a)$.
6. From $T_6 d(\omega_1, ..., \omega_\nu, \omega; \omega, s)$ follows that
   if not $[d(\omega_1, ..., \omega_\nu; s)$ iff $d(\omega_1, ..., \omega_\nu; a(\omega, s)]$ then $\text{Prohibited}_{\omega,s}(a)$.
7. From $T_7 d(\omega_1, ..., \omega_\nu, \omega; \omega, s)$ follows that
   if $d(\omega_1, ..., \omega_\nu; a(\omega, s))$ then $\text{Prohibited}_{\omega,s}(a)$.

These principles can be used to define a deontic structure-operator Δ such that to each agent ω in a state s is assigned the set of feasible acts a that are not eliminated as $\text{Prohibited}_{\omega,s}(a)$ according to the rules (1)-(7) above. Since

$\text{Prohibited}_{\omega,s}(a)$ is equivalent to $\neg\text{Permissible}_{\omega,s}(a)$.

it follows that

$\Delta(\omega, s) = \{\text{Permissible}_{\omega,s}(a) : a \in A\}$.

Note that at the outset, all feasible acts are permissible. The basic idea is that we eliminate elements from the set of permissible acts for ω in s using the norms and sentences expressing what holds for the agents with respect to grounds in the norms.

The method used for representing norms in an architecture for norm-regulated Mas can be of importance for the effectiveness of the architecture. Here a few examples of what can be regarded as desiderata for a norm-representation method are mentioned.

1. The system of norms is depicted in a lucid, concise and effective way.
2. Changes and extensions of the normative system are easily described.
3. The normative system can be divided in different parts which can be changed independently.
4. The multi-agent system can by itself change the normative system wholly or partially.

The last item in the list may deserve a comment. It is often difficult to predict the effect of a normative system for a Mas or the effect of a change of norms. It is therefore desirable that the Mas can by itself evaluate the effect of the normative system and compare the result with other normative systems that it changes to. The result can be a kind of evolution of normative systems obtained by machine learning.

In Odelstad & Boman (2004), the *npcis*-model was used for representing normative systems, which resulted in an opportunity to test some aspects of this kind of representation in the area of multiagent-systems.

8.4 Prolog implementation of norm-regulated DALMAS

In Hjelmblom (2008), an implementation in Prolog of the theory of a norm-regulated DALMAS is presented. The algebraic theory is instrumentalized through an executable logic program. Important issues in the transition from a set-theoretical description to a Prolog implementation are discussed. Results include a general-level Prolog implementation, which may be freely used to implement specific systems.

The Prolog implementation gives a procedural semantics to the algebraic theory, see Lloyd (1987). Running the Prolog program has not only pedagogical value, but can aid understanding of the implications of changing parts of the underlying theory. The fact that the Prolog program runs without notably long response time also testifies, albeit informally, to the acceptable computational complexity of the canonical model. Any domain-specific model created with Hjelmblom's Prolog implementation can have its computational complexity analysed more formally through algorithmic analysis if necessary (see Purdom Jr. & Brown, 1985).

8.5 Norms and forest cleaning

Forest management treatments presuppose, in a state of incomplete information, principles for choosing those trees that ought to be taken away and those that shall be left standing. In this section, which is a report on a work in progress carried out in cooperation with Ulla Ahonen-Jonnarth, the question is raised whether those principles can be structured as a combination of a normative system and a utility function. Of special interest is the possibility to evaluate the efficiency of the normative system and the utility function and, furthermore, suggest improvements of them.[21]

In the forest industry there is an increasing interest in the automation of forest management treatments, perhaps with the ultimate goal that autonomous robots will be able to do a substantial part of such work. But before robots of this kind can be constructed many difficult problems must be solved, for example how the robots will perceive the environment and how they will transport themselves. But there are also decision-making problems involved. Three important kinds of forest management treatments are cleaning, thinning and harvesting, and they all require methods or principles for making decisions about which trees shall be removed and which will be left standing. Such "remove-decisions" must be made on-line with information based only on the robot's nearest vicinity and about that part of the stand already cleared. The treatment cannot be evaluated

[21] This section is based on Ahonen-Jonnarth & Odelstad (2005), Ahonen-Jonnarth & Odelstad (2006) and Odelstad (2007).

until the actual stand is completely cleared. Testing and evaluating principles for remove-decisions by field experiments is expensive and time-consuming. It is therefore an interesting question wether evaluating experiments could be made *in silico*, i.e. through simulation.

In Ahonen-Jonnarth & Odelstad (2005), a platform for simulation of young forest stands is presented. Given field data of a special type of young forest, for example a 10-year-old, somewhat damp, spruce forest at 200 meters above sea level in the middle of Sweden, it is possible to simulate different stands of this type of forest. Field data of a few different types of young forests has so far been used for simulation. As a base for the simulation of different stands of the same forest type, it is of course also possible to use man-made, artificial data, or to assign values to the parameters that govern the simulation.

One of the goals of our present work on automation of forest cleaning is to formulate different principles for making the remove-decisions, test the principles in simulated forests of different types and evaluate and compare the results. We are especially interested in the possibility that, given a method for evaluating the result of cleaning, the system can improve the decision-making principles and even suggest new ones on the basis of machine learning. How the principles for the remove-decisions ought to be formally represented seems to be a complicated question. One possibility we want to investigate is to use norm-regulated Dalmas as the architecture for a cleaning agent. At this preliminary stage, a cleaning agent is regarded as "a solitary being" and, hence, a cleaning Dalmas is a one-agent-system (thus more correctly a Daloas), but we will here regard a one-agent-system as a degenerated MAS. But at a later stage, more then one agent may be involved, for example can 'nature' be regarded as an agent or can individual trees be regarded as agents. The last mentioned alternative is especially interesting if the growth of a forest stand is incorporated in the simulation.

A Dalmas can achieve the cleaning-decisions for a stand p in the following way. The stand is divided into n different areas. A state for the system is the stand with i areas cleaned, where $1 \leq i \leq n$, and a specification of what area to clean next. The initial state is the stand with 0 areas cleaned and the final state is the state with n areas cleaned. Let each area be denoted by a unique number between 1 and n, and let S_i be the ith state. C_i denotes the set of cleaned areas and U_i the set of uncleaned areas in S_i. Thus, $C_i \cup U_i = \{1, 2, \ldots, n\}$ and $C_i \cap U_i = \emptyset$. C_i contains i numbers and U_i contains $n-i$ numbers. $S_i = \langle C_i, U_i, j\rangle$ where j is the area which will be cleaned next, i.e. $j \in U_i$ and $S_{i+1} = \langle C_i \cup \{j\}, U_i \backslash \{j\}, k\rangle$ for some $k \in U_i \backslash \{j\}$.

A few examples of possible norms regulating a cleaning Dalmas are given below:

(a) If there is only one undamaged tree in the area to be cleaned with a diameter within the desirable range, then this tree shall be saved.

(b) If there is at least one undamaged tree in the area to be cleaned with a diameter within the desirable range, then a damaged tree with a diameter below the desirable range may be taken away.

(c) If, in the area to be cleaned, a tree t is damaged and is closer than 0.5m to an undamaged tree with a diameter within the desirable range and with distances to other undamaged trees larger than 0.5m, then t may not be saved.

In many situations, the norms of a Dalmas do not determine the action to be taken in each state, but utility considerations are also necessary. Given a utility function we can search for the optimal way of cleaning the actual area, on the assumption that the cleaning satisfies the given norms.

For the possibility of using norms in the automation of forest cleaning in the way outlined above, it may be an important issue whether the cleaning system can optimize the system of norms regulating its remove-decisions. This is a special case of a more general problem: Suppose that $\mathfrak{D}$ is a Dalmas, where the agents cooperate to solve a problem. Which normative system will lead to the most effective behavior of the system? It is desirable that $\mathfrak{D}$ itself could determine the optimal normative system for the task in question. Given a set of grounds and a set of consequences, which together constitute the vocabulary of the system, $\mathfrak{D}$ can test all possible sets of minimal norms (in many cases satisfying certain constraints, for example represented by intermediaries). If there is a function for evaluating the result of a run of $\mathfrak{D}$, then different normative systems can be compared and the best system can be chosen. A change of vocabulary corresponds to a "mutation" among normative systems and can lead to dramatic changes in the effectiveness. Note that, in principle, the evaluation function can be very complicated, for example it can be multi-dimensional.

Acknowledgement
Heartfelt thanks to Lars Lindahl, Magnus Boman, Ulla Ahonen-Jonnarth and Magnus Hjelmblom for fruitful cooperations and stimulating discussions. Financial support was given by Gävle University, the KK-foundation and the Swedish Research Council.

9 References

Ahonen-Jonnarth, U, Odelstad, J. (2005) Simulation of cleaning of young forest stands. *Reports from Creativ Media Lab,* 2005:2. University of Gävle.

Ahonen-Jonnarth, U. & Odelstad, J. (2006) Evaluation of Simulations with Conflicting Goals with Application to Cleaning of Young Forest Stands. *Proceedings of ISC 2006* (Fourth Annual International Industrial Simulation Conference), Palermo, Italy, June 5-7, 2006.

Alchourrón, C.E. & Bulygin, E. (1971) *Normative Systems.* Springer, Wien.

Dummett, M. (1973) *Frege: Philosophy of Language*, Duckworth, London.

Einstein, A. (1936) Physics and Reality. *The Journal of the Franklin Institute*, Vol. 221, NO 3, March 1936. Reprinted in A. Einstein: *Ideas and opinions*. Souvenir Press 1973.

Einstein, A. (1949) Reply to Criticisms. In P. A. Schilpp (ed.) *Albert Einstein: philosopher-scientist.* Evanston, Ill. : Library of Living Philosophers.

Einstein, A. (1973) *Ideas and opinions.* Souvenir Press.

Ekelöf, P.-O. (1945) Juridisk slutledning och terminologi, *Tidsskrift for Rettsvitenskap,* **58**, pp. 211 ff.

Gentzen, G. (1934). Untersuchungen über das logische Schließen, I. *Mathematische Zeitschrift,* **39**, 176-210.

Halldén, S. (1978) Teckenrelationen och språkreglernas juridik. In *En Filosofibok Tillägnad Anders Wedberg.* Stockholm.

Hare, R. (1989) *Essays in Ethical Theory.* Oxford University Press, Oxford.

Heath, P.L. (1967) Concept. In P. Edwards (ed.) *The Encyclopedia of Philosophy*, Macmillan, New York.

Hedenius, I. (1941) *Om rätt och moral.* Tidens förlag, Stockholm. Second ed. Wahlström & Widstrand, Stockholm, 1965.

Hjelmblom, M. (2008). *Deontic Action-Logic Multi-Agent Systems in Prolog.* Thesis work for degree of Master of Science in Computing Science, Uppsala University, FoU-rapport Nr 30, University of Gävle.

Hohfeld, W.N. (1923) *Fundamental Legal Conceptions as Applied in Judicial Reasoning and Other Legal Essays* (ed. W.W. Cook), Yale University Press, New Haven.

Jones, A.J.I. & Sergot, M.J. (1993) On the Characterisation of Law and Computer Systems: The Normative Systems Perspective. In J-J.Ch. Meyer & R.J. Wieringa, (eds.) *Deontic Logic in Computer Science: Normative System Specification.* Wiley.

Jones, A.J.I. & Sergot, M.J.(1996) A Formal Characterisation of Institutionalised Power. *Journal of the IGPL* 4 (3):429-445. Reprinted in Valdés, E.G. et al.(eds.) *Normative Systems in Legal and Moral Theory. Festschrift for Carlos E. Alchourrón and Eugenio Bulygin.* Duncker & Humboldt, Berlin, 1997, pp. 349-367.

Kanger, S. (1957) *New Foundations for Ethical Theory.* Part 1. Stockholm. Reprinted in R. Hilpinen (ed.) *Deontic Logic: Introductory and Systematic Readings*, pp. 36-58. Dordrecht, 1971.

Krogh, C. (1995) The Rights of Agents. *Intelligent Agents II,* IJCAI'95 Workshop (ATAL), Springer.

Krogh, C. & Herrestad, H. (1999) Hohfeld in Cyberspace and Other Applications of Normative Reasoning in Agent Technology", *Artificial Intelligence and Law* 7: 81-96.

Lindahl, L. (1968) Om tysta löften inom civilrätten. In *Sanning, Dikt, Tro: Till Ingemar Hedenius.* Bonniers, Stockholm.

Lindahl, L. (1977) *Position and Change. A Study in Law and Logic.* Reidel, Dordrecht.

Lindahl, L. (1985) Definitioner, begreppsanalys och mellanbegrepp i juridiken. In *Rationalitet och Empiri i Rättsvetenskapen,* pp. 37-52. Juridiska Fakultetens i Stockholm skriftserien, nr 6. Stockholm.

Lindahl, L. (1994) Stig Kanger's Theory of Rights. In D. Prawitz, B. Skyrms & D. Westerståhl (eds.) *Logic, Methodology and Philosophy of Science IX,* pp. 889-911 Elsevier.

Lindahl, L. (1997) Norms, Meaning Postulates, and Legal Predicates. In Valdés et al. (eds) *Normative Systems in Legal and Moral Theory. Festschrift for Carlos E. Alchourrón and Eugenio Bulygin,* pp. 293-307. Duncker & Humblot, Berlin.

Lindahl, L. (2000) Deskription och normativitet: De juridiska begreppens Janusansikte. In Numhauser-Henning, A. (ed.) *Normativa perspektiv. Festskrift till Anna Christensen.* Lund.

Lindahl, L. (2003) Operative and Justificatory Grounds in Legal Argumentation. In K. Segerberg & R. Sliwinsky (eds.) *Logic, law, morality: thirteen essays in practical philosophy in honour of Lennart Åqvist,* pp. 111-126. Uppsala philosophical studies 51, Department of Philosophy, Uppsala.

Lindahl, L. (2004) Deduction and Justification in the Law. The Role of Legal Terms and Concepts. *Ratio Juris* **17**: 182 - 202.

Lindahl, L. (2005) Hohfeld Relations and Spielraum for Action. In C. Dahlman (ed.) *Studier i rättsekonomi: festskrift till Ingemar Ståhl,* pp. 121-150. Studentlitteratur, Lund.

Lindahl, L. & Odelstad, J. (1996) Grounds and consequences in conceptual systems. In S. Lindström, R. Sliwinski och J. Österberg (eds.) *Odds and Ends. Philosophical Essays Dedicated to Wlodek Rabinowicz.* Uppsala Philosophical Studies 45, Department of Philosophy, Uppsala.

Lindahl, L. & Odelstad, J. (1999a) Intermediate Concepts as Couplings of Conceptual Structures. In H. Prakken, & P. McNamara (ed*.) Norms, Logics and Information Systems. New Studies on Deontic Logic and Computer Science.* IOS Press, Amsterdam.

Lindahl, L. & Odelstad, J. (1999b). Normative systems: core and amplifications. In R. Sliwinski (ed.) *Philosophical Crumbs. Essays Dedicated to Ann-Mari Henschen-Dahlquist.* Uppsala Philosophical Studies 49. Department of Philosophy, Uppsala.

Lindahl, L. & Odelstad, J. (2000). An algebraic analysis of normative systems. *Ratio Juris* 13: 261-278.

Lindahl, L. & Odelstad, J. (2003) Normative Systems and Their Revision: An Algebraic Approach. *Artificial Intelligence and Law,* 11: 81-104.

Lindahl, L. & Odelstad, J. (2004) Normative Positions within an Algebraic Approach to Normative Systems. *Journal Of Applied Logic,* 2: 63-91.

Lindahl, L. & Odelstad, J. (2006a). Intermediate Concepts in Normative Systems. In L. Goble & J-J.Ch. Meyer (eds.) *Deontic Logic and Artificial Normative Systems.* (DEON 2006). LNAI 4048, pp. 187-200. Springer-Verlag.

Lindahl, L. & Odelstad, J. (2006b). Open and Closed Intermediaries in Normative Systems. In T.M. van Engers (ed.) *Legal Knowledge and Information Systems.* (Jurix 2006). IOS Press, Amsterdam.

Lindahl, L. & Odelstad, J. (2008a) Intermediaries and Intervenients in Normative Systems. *Journal of Applied Logic* 6: 229-250.

Lindahl, L. & Odelstad, J. (2008b) Strata of Intervenient Concepts in Normative Systems. In R. van der Meyden & L. van der Torre (eds.): *DEON 2008,* LNAI 5076, pp. 203-217. Springer-Verlag, Berlin Heidelberg.

Lloyd J.W. (1987) *Foundations of Logic Programming.* Second, Extended Edition. Springer-Verlag.

Luger, G.F. (2002) *Artificial Intelligence. Structures and Strategies for Complex Problem Solving,* 4th ed. Addison-Wesly.

Makinson, D. (1986) On the Formal Representation of Right Relations. *Journal of Philosophical Logic,* **15**, 403-425.

Makinson, D. & van der Torre, L. (2000). Input-output Logics. *Journal of Philosophical Logic,* **29**, 383-408.

Makinson, D. & van der Torre, L. (2003) What is input/output logic? In *Foundations of the Formal Sciences II: Applications of Mathematical Logic in Philosophy and Linguistics,* In: *Trends in Logic*, vol. 17, Kluwer, Dordrecht.

Nagel, E. (1961) *The Structure of Science.* Routledge, London.

Negnevitsky, M. (2005) *Artificial Intelligence. A Guide to Intelligent systems.* Second Edition. Addison-Wesley.

Odelstad, J. (1989) *Om den metodologiska subjektivismen.* Lecture delivered in the higher seminar of theoretical philosophy, Uppsala University, April 1989. (Unpublished manuscript in Swedish.)

Odelstad, J. (1992) *Invariance and Structural Dependence.* Lecture notes in Economics and Mathematical Systems 380. Springer-Verlag.

Odelstad, J. (2002a) Norms for Multi-Agent Systems - The Representation Problem. In J. Bubenko, J. & B. Wangler (eds) *Promote IT 2002.* Proceedings of the Second Conference for the Promotion of Research in IT. Skövde.

Odelstad, J. (2002b) *Artificial Agents and Norms.* Department of Computer and Systems Sciences, SU & KTH, Masters' series, No. 02-80-DSV-SU, Stockholm.

Odelstad, J. (2002c) *Intresseavvägning. En beslutsfilosofisk studie med tillämpning på planering.* Thales, Stockholm.

Odelstad, J. (2003) An Abstract Architecture for Norm-Regulated Agents. In *Promote IT 2003.* Proceedings of the Third Conference for the Promotion of Research in IT, Visby.

Odelstad, J. (2007) Agents, Norms and Forest Cleaning. In G. Boella, L. van der Torre & H. Verhagen (eds.): *Normative Multi-Agent Systems,* Dagstuhl Seminar Proceedings 07122, ISSN 1862 – 4405.
URL http://drops.dagstuhl.de/portals/index.php?semnr=07122.

Odelstad, J. (2008a) Likvärdigt arbete - en logisk och rättsfilosofisk analys. (‘Work of equal value–a logical and philosophical analysis’, in Swedish, 40 pp.) To be published.

Odelstad, J. (2008b) *Many-Sorted Implicative Conceptual Systems.* DSV Report series No. 08-012. Royal Institute of Technology, Stockholm.

Odelstad, J. & Boman, M. (2004) Algebras for Agent Norm-Regulation. *Annals of Mathematics and Artificial Intelligence,* 42: 141-166.

Odelstad, J. & Lindahl, L. (1998) *Conceptual Structures as Boolean Orderings.* In L. Lindahl, J. Odelstad & R. Sliwinski (eds.) *Not Without Cause. Philosophical Essays Dedicated to Paul Needham.* Uppsala Philosophical Studies 48, Department of Philosophy, Uppsala.

Odelstad, J. & Lindahl, L. (2000) Normative Systems Represented by Boolean Quasi-Orderings. *Nordic Journal of Philosophical Logic,* **5**, 161-174.

Odelstad, J. & Lindahl, L. (2002) The Role of Connections as Minimal Norms in Normative Systems. In T. Bench-Capon, A. Daskalopulu & R. Winkels (eds.) *Legal Knowledge and Information Systems.* (Jurix 2002) IOS Press, Amsterdam.

Olsson, J. (2006) *Normsystem för Wastecollectors-systemet.* Candidate Thesis, University of Gävle.

Purdom Jr., P.W. & Brown, C.A. (1985) *The Analysis of Algorithms.* Holt, Rinehart & Winston, London.

Ross, A. (1951) Tû-Tû. In O.A. Borum & K. Illum (eds.) *Festskrift til Henry Ussing.* København: Juristforbundet. (English translation as Ross (1956-57)).

Ross, A. (1956-57) Tû-Tû. *Harvard Law Review,* **70**, 812-825. (English translation of Ross (1951)).

Sergot, M. (1999) Normative Positions. In P. McNamara & H. Prakken (eds.): *Norms, Logics and Information Systems,* pp. 289-308. IOS Press, Amsterdam.

Sergot, M. (2001) A Computational Theory of Normative Positions *ACM Transactions on Computational Logic,* **2** , 581-622.

Suppes, P. (1957) *Introduction to Logic.* Van Nostrand, New York.

von Wright, G.H. (1951) Deontic Logic, *Mind,* **60**, 1-15.

Wedberg, A. (1951). Some problems in the logical analysis of legal science. *Theoria,* **17**, 246-275.

Argumentation based Resolution of Conflicts Between Desires and Normative Goals

Sanjay Modgil and Michael Luck

Department of Computer Science, Kings College London

Abstract. Norms represent what ought to be done, and their fulfillment can be seen as benefiting the overall system, society or organisation. However, individual agent goals (desire) may conflict with system norms. If a decision to comply with a norm is determined exclusively by an agent or, conversely, if norms are rigidly enforced, then system performance may be degraded, and individual agent goals may be inappropriately obstructed. To prevent such deleterious effects we propose a general framework for argumentation-based resolution of conflicts amongst desires and norms. In this framework, arguments for and against compliance are arguments justifying rewards, respectively punishments, exacted by 'enforcing' agents. The arguments are evaluated in a recent extension to Dung's abstract argumentation framework, in order that the agents can engage in *metalevel* argumentation as to whether the rewards and punishments have the required motivational force. We provide an example instantiation of the framework based on a logic programming formalism.

1 Introduction

Requirements for conflict resolution arise in open multi-agent systems in which goals of individual agents conflict with norms imposed by the system to regulate individual agent behaviour. If the decision to comply with a norm is determined exclusively by an individual, then system performance may be degraded. Hence, institutional or social pressure to comply may be brought about by *system agents* exacting punishments and grants rewards [17, 11]. This may be appropriate for closed static systems, but compromises the flexibility of dynamic open systems in which rigid enforcement of norms may lead to both unwarranted obstruction of agent goals and degraded system performance. For example, an agent's goal may be obstructed by enforcing compliance with a norm that is justified by system-held beliefs about the context. However, these beliefs may be erroneous. In addition, it may not always be able to anticipate at design time, contexts in which compliance with norms does or does not coincide with the best interests of the system, and when enforcement mechanisms have insufficient motivational force. In such cases, an agent might appeal to higher level *motivations* [9], arguing that in pursuing its own goal it is indeed acting in the interests of the system as a whole, or that exacted punishments (or rewards) for non-compliance (or compliance) are outweighed by the benefits of pursing its own goal.

In this paper we propose a general argumentation-based framework that evaluates arguments for and against compliance with norms, in order to prevent unwarranted obstruction of individual goals and degraded system performance. As in [11, 6], norms

Dagstuhl Seminar Proceedings 09121
Normative Multi-Agent Systems
http://drops.dagstuhl.de/opus/volltexte/2009/1912

are interpreted as *system goals* that individual agents are required to realise, and that may conflict with the *individual goals* or *desires* of an agent. Punishments and rewards are the individual goals of system agents responsible for enforcement. In general, an argument for a goal justifies realisation of that goal based on beliefs that are themselves the outcome of argumentation based reasoning about what is the case. An argument for a system goal may then mutually attack an argument for a conflicting individual goal, and arguments for punishment and reward goals attack the argument for an individual goal. It is the success of these attacks that determines which of the arguments prevail and thus whether or not there is a reasoned case for compliance[1]. In general, an attack succeeds as a defeat if the attacked argument is not stronger than or *preferred* to its attacker [1]. As in [4], preferences may be derived from a relative ordering on the values that the arguments promote. In this paper, preferences among arguments for goals are similarly evaluated. For example, a 'reward argument' will successfully attack (defeat) an argument for an individual goal if an agent is persuaded that the reward is of greater utility to it than the individual goal it is required to abandon in favour of compliance with the system goal. The proposed framework will thus need to account for:

1. **Social mechanisms for enforcing compliance**: An agent Ag's argument for an individual goal g may be attacked by arguments for the (punishment and reward) goals $g', \ldots$ of *other* agents, where the attacks are not based on direct conflicts between g and $g', \ldots$. For example, a reward (punishment) may facilitate (hinder) some other goal that Ag is already committed to realising.
2. **Motivational argumentation**: Flexible and adaptive agents need to engage in motivational argumentation over the respective merits of goals. Hence, argumentation frameworks in which preferences [1] and value orderings [4] on arguments are 'given', and not themselves subject to reasoning, do not suffice. Rather, there is a requirement for argumentation based reasoning over the preferences themselves.

Existing work addresses argumentation-based resolution of conflicts among goals ([2], [10], [16]), and [16] explicitly considers conflicts between individual goals and norms. However, no existing work accounts for social mechanisms, whereby an agent's decision as to which goals to pursue is influenced by other agents' goals. Only [10] accounts for argumentation over preferences, but does so in the object level logic programming language, whereby rules express priorities over other rules. In this paper, we aim at an abstract framework in which preferences are not restricted to rule priorities, but can account for any criteria for valuating argument strength, including those that relate to the argument as a whole (e.g., as in [4]). We therefore make use of a recent extension to Dung's seminal abstract argumentation semantics [8]. In a Dung framework, arguments are related by a binary conflict-based relation, and the winning (justified) arguments under different extensional semantics are evaluated. The underlying logic, and definition of the logic's constructed arguments and conflict relation, is left unspecified, enabling instantiation by various logical formalisms. Dung's semantics thus serves as a general framework capturing various species of non-monotonic reasoning [5], and, more generally for conflict resolution. Hence, approaches to argumentation based agent

[1] In philosophical parlance we are adopting an *externalist* rather than *internalist* view, where the latter consider norms to be intrinsically motivating.

reasoning often conform to these semantics, whereby an agent's inferences (e.g. denoting beliefs or goals) can be defined in terms of the claims of the justified arguments constructed from the underlying theory (an argument essentially being a proof of a candidate inference — the argument's claim — in the underlying logic). In [12, 13], Dung's semantics have been extended to accommodate arguments that *express preferences between other arguments*, where no assumption is made as to how these preferences are defined in the instantiating formalism.

In Section 2 we review the extended semantics described in [12, 13]. The main contributions of this paper are then described in Sections 3, 4 and 5. In Section 3 we describe a general framework for argumentation based resolution of conflicts between system norms and agent goals. Specifically, we combine the extended argumentation semantics with the normative model of [11] in which compliance with norms is enforced through punishments and rewards modelled as the goals of enforcement agents. The framework thus provides for social mechanisms for enforcing compliance, and motivational argumentation. Section 4 then describes a logic programming instantiation of the general framework. Section 5 illustrates the instantiation with an extended example. Finally, Section 6 concludes with a discussion of related and future work.

2 Extended Argumentation Frameworks for Agent Reasoning

2.1 Dung's Argumentation Framework

A Dung argumentation framework is a tuple $(Args, \mathcal{R})$ where $\mathcal{R} \subseteq (Args \times Args)$ can denote either an 'attack' or 'defeat' relation, and where the latter can be understood as an attack that succeeds given the available preference information. An argument $A \in Args$ is defined as acceptable w.r.t. some $S \subseteq Args$, if for every B such that $(B, A) \in \mathcal{R}$, there exists a $C \in S$ such that $(C, B) \in \mathcal{R}$. Intuitively, C 'reinstates' A. Dung then defines the acceptable extensions of $(Args, \mathcal{R})$ under different extensional semantics. In this paper we focus on the admissible and preferred semantics. Letting $S \subseteq Args$ be conflict free if no two arguments in S are related by $\mathcal{R}$, then:

Definition 1. *Let $S \subseteq Args$ be a conflict free set.*

- *S is admissible iff each argument in S is acceptable w.r.t. S*
- *S is a preferred extension iff S is a set inclusion maximal admissible extension*

An argument is said to be justified if it belongs to all preferred extensions of a framework.

2.2 Motivating Extended Argumentation Frameworks

We now motivate extending Dung's framework with the following example (that will be referred to later in Section 3).

Example 1. Consider two agents $Ag1$ and $Ag2$ exchanging arguments A, $B\dots$ about the weather forecast for Hawaii.

$Ag2$: “According to the BBC it will be cool in Hawaii” = A
$Ag1$: “According to CNN it will be hot in Hawaii” = B
$Ag2$: “But the BBC are more trustworthy than CNN” = C
$Ag1$: “However, statistics show CNN are more accurate than the BBC” = D
$Ag1$: “And a statistical comparison is more rigorous and rational than basing a comparison on your instincts about their relative trustworthiness” = E

Arguments A and B symmetrically attack, i.e., (A, B),$(B, A) \in \mathcal{R}$. $\{A\}$ and $\{B\}$ are the preferred extensions, and so neither argument is justified. We then have argument C claiming that A is preferred to B. Hence B does not successfully attack (defeat) A, but A does defeat B. Intuitively, C is an argument for A’s repulsion of, or defence against, B’s attack on A; i.e., C attacks B***’s attack on*** A ($\Delta2$ in Figure 1a)) so that B’s attack on A does not succeed as a defeat. B’s attack on A is, as it were, cancelled out, and we are left with A defeating B. Evaluating the preferred extensions on the basis of $\mathcal{R}$ denoting the defeat relation, we now have the single preferred extension containing A. Now, given D claiming a preference for B over A and so attacking A’s attack on B, neither defeat the other and so once again we have two preferred extensions. Since C and D claim contradictory preferences they attack each other ($\Delta3$). These attacks can themselves be subject to attacks in order to determine the defeat relation between C and D and, in so doing A and B. E attacks the attack from C to D ($\Delta4$), so that D defeats C, B defeats A, and $Ag1$’s argument that it will be hot in Hawaii is now justified.

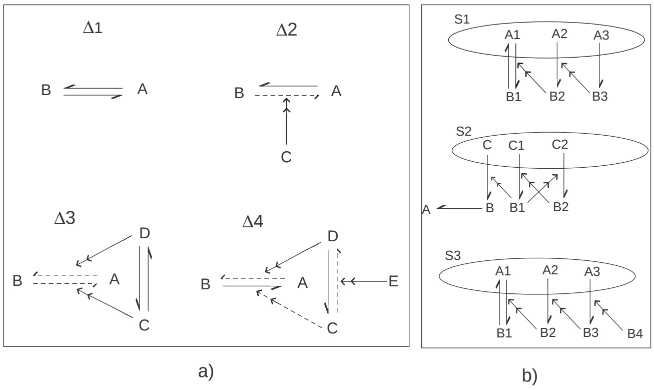


Fig. 1. Motivating *EAF*s

2.3 Defining Extended Argumentation Frameworks

Example 1 illustrates requirements for arguments attacking attacks. Hence, as in [12, 13], an *Extended Argumentation Framework* is defined as follows:

Definition 2. *An* Extended Argumentation Framework *(*EAF*) is a tuple ($Args$, $\mathcal{R}$, $\mathcal{D}$) such that $Args$ is a set of arguments, and:*

- $\mathcal{R} \subseteq Args \times Args$
- $\mathcal{D} \subseteq (Args \times \mathcal{R})$
- *If* $(C, (B, A))$, $(D, (A, B)) \in \mathcal{D}$ *then* (C, D), $(D, C) \in \mathcal{R}$

Notation 1 We may write $A \rightharpoonup B$ to denote $(A, B) \in \mathcal{R}$. If in addition $(B, A) \in \mathcal{R}$, then $A \rightleftharpoons B$. Also, $D \twoheadrightarrow (A \rightharpoonup B)$ denotes $(D, (A, B)) \in \mathcal{D}$

The defeat relation is now parameterised w.r.t. some set S of arguments. This accounts for an attack's success as a defeat being relative to preference arguments already accepted in some set S, rather than relative to some externally given preference ordering.

Definition 3. *A defeats$_S$ B, denoted by $A \rightarrow^S B$, iff $(A, B) \in \mathcal{R}$ and $\neg \exists D \in S$ s.t. $(D,(A, B)) \in \mathcal{D}$.*

Referring to Example 1, A defeats$_\emptyset$ B but A does not defeat$_{\{D\}}$ B. The notion of a conflict free set S of arguments is now defined. Notice that it may be that an argument A asymmetrically attacks an argument B, so that given $D \twoheadrightarrow (A \rightharpoonup B)$, neither A nor B defeat$_S$ each other if $D \in S$. This means that both A and B may be accepted together in the same extension (where any extension is required to be conflict free). For example, if B is an argument for an action, and A claims that (for example) the action is too costly, it may be that an agent decides to execute the action while accepting that it is expensive (in value based argumentation [4], D is an argument claiming that the value promoted by B's action is greater than A's value of 'cost'). In the following section we will show that such *preference dependent asymmetric* attacks are also appropriate when resolving conflicts between norms and desires.

Definition 4. *S is conflict free iff $\forall A, B \in S$: if $(A, B) \in \mathcal{R}$ then $(B, A) \notin \mathcal{R}$, and $\exists D \in S$ s.t. $(D,(A, B)) \in \mathcal{D}$.*

The definition of acceptability of an argument A w.r.t. a set S for an *EAF* is motivated in some detail in [12, 13]. It references the notion of a *reinstatement set* for a defeat, in order that an intuitive requirement on what it means for an argument to be acceptable w.r.t. an admissible set S of arguments is satisfied: *if A is acceptable with respect to S, then $S \cup \{A\}$ is admissible* (referred to as the fundamental lemma in Dung [8]).

Definition 5. *Let $S \subseteq Args$ in ($Args$, $\mathcal{R}$, $\mathcal{D}$). Let R_S = $\{X_1 \rightarrow^S Y_1, \ldots, X_n \rightarrow^S Y_n\}$ where for $i = 1 \ldots n$, $X_i \in S$. Then R_S is a reinstatement set for $C \rightarrow^S B$, iff:*
- • *$C \rightarrow^S B \in R_S$, and*
- • *$\forall X \rightarrow^S Y \in R_S$, $\forall Y'$ s.t. $(Y',(X, Y)) \in \mathcal{D}$, $\exists X' \rightarrow^S Y' \in R_S$*

Definition 6. *Let* $(Args, \mathcal{R}, \mathcal{D})$ *be an* EAF. $A \in Args$ *is acceptable w.r.t.* $S \subseteq Args$ *iff* $\forall B \in Args$ *s.t.* $B \rightarrow^S A$, $\exists C \in S$ *s.t.* $C \rightarrow^S B$ *and there is a* reinstatement set *for* $C \rightarrow^S B$.

In Figure 1b), $A1$ is acceptable w.r.t. $S1$. We have that $B1 \rightarrow^{S1} A1$, and $A1 \rightarrow^{S1} B1$. The latter defeat is itself *challenged* by $B2$. However, $A2 \rightarrow^{S1} B2$, which in turn is challenged by $B3$. But then, $A3 \rightarrow^{S1} B3$. We have the reinstatement set $\{A1 \rightarrow^{S1} B1, A2 \rightarrow^{S1} B2, A3 \rightarrow^{S1} B3\}$ for $A1 \rightarrow^{S1} B1$. Also, A is acceptable w.r.t. $S2$ given the reinstatement set $\{C \rightarrow^{S2} B, C1 \rightarrow^{S2} B1, C2 \rightarrow^{S2} B2\}$ for $C \rightarrow^{S2} B$. Finally $A1$ is not acceptable w.r.t $S3$ since no argument in $S3$ defeats$_{S3}$ $B4$.

Admissible and preferred semantics for *EAF*s are now given by Definition 1, where conflict free is defined as in Definition 4. (In [12, 13], the complete, stable and grounded semantics are similarly defined for *EAF*s, i.e., in the same way as for Dung frameworks). Referring to Example 1, $\{B, D, E\}$ is the single preferred extension. In [12, 13] we show that *EAF*s inherit many of the fundamental results holding for extensions of a Dung framework. This suggests that much of the work building on Dung's framework can readily be reformulated for *EAF*s, including work on argument game proof theories and dialogue frameworks. In particular, Dung's fundamental lemma is satisfied, implying that the set of all admissible extensions of an *EAF* form a complete partial order w.r.t. set inclusion, and so for each admissible S there exists a preferred extension S' such that $S \subseteq S'$.

To conclude, the extended semantics accommodates arguments that express preferences between other arguments, while preserving the abstraction of a Dung framework; no commitments are made to how preferences are defined in the instantiating logical formalism. We now make use of the extended semantics in a framework for conflict resolution in normative systems, and show that the ability to engage in argumentation based reasoning *about*, as well as *with*, defeasible and possibly conflicting preference information, provides for agent flexibility and adaptability.

3 A Framework for Conflict Resolution in Normative Systems

This section describes a framework in which agents engage in dialogues to decide which amongst conflicting desire based and normative goals are to be pursued. Agent submit arguments for goals, where these arguments attack each other, and then engage in motivational argumentation over the relative utility of states in which the goals are realised. This equates to arguing over preferences between arguments, and so which attacks succeeds as defeats. The arguments and attacks defined in the course of a dialogue thus instantiate an *EAF*, and the goals to be pursued are those claimed by the justified arguments of the *EAF*. Note that the agents also argue over the beliefs justifying adoption of goals. In this way, the agents are first required to agree that the goal being proposed for adoption is indeed warranted by what is believed to be the case. Section 3.1 first sets out some general assumptions about the kinds of agents modelled by the framework, and the dialogues these agents participate in. Section 3.2 then describes how conflicts between individual agent goals and system norms are resolved through argumentation based dialogues over beliefs, and goals proposed by individual agents and agents acting on behalf of the system.

3.1 Agents and Dialogues

The proposed framework abstracts from the logics for agent reasoning, assuming only *BDI* type agents (e.g. those instantiating the *BOID* architecture [7]) and a declarative interpretation of goals as beliefs holding in some future state. Each agent has a belief base consisting of facts and rules, and a goal base containing rules for deriving goals. From amongst all the goals that are derivable, those that an agent commits to realising are referred to as intentions. An intention persists in an agent's intention base until such a time as it is realised by a plan (the agent's planning component is not modelled here).

As in [11]'s model of normative multi-agent systems, four types of goal are distinguished. Individual agent goals, which we refer to here as *desires*, may conflict with *normative goals*. For example, an agent $Ag1$'s desire to stay on Waikiki beach in Hawaii, may conflict with the normative goal of staying in a cheap hotel. $Ag1$ may decide to comply or not comply with the norm, based on rewards and punishments exacted by system agents (specifically *enforcement* agents). Rewards and punishments are also individual goals of enforcement agents, but are punishment, respectively reward goals, from the perspective of the agent being punished, respectively rewarded. *Punishment goals* hinder the punished agent's intentions if that agent decides not to comply with the norm. For example, a punishment may be to deny the funding that $Ag1$ needs to fulfill its intention to visit Leipzig for a meeting. *Reward goals* benefit the achievement of the rewarded agent's intentions if it decides to comply. For example, a reward for an agent who intends to have a laptop, may be to provide the agent with a laptop.

In general, goals are derived by rules whose antecedents refer to what the agent believes and its current intentions. Extending the scenario described in Example 1, suppose agent $Ag1$ believes it will be hot in Hawaii, and it intends to attend a conference in Hawaii. Then it derives the desire to stay on Waikiki beach. The goals of system agents are derived in the same way, and may additionally refer to the intentions of other agents. For example, if $Ag1$ intends to attend a conference, then the normative goal of staying in a cheap hotel is derived (in either $Ag1$'s goal base or the goal base of a system agent responsible for informing other agents of their obligations). An enforcement agent Ag_P may derive the punishment goal of denying $Ag1$ the funding for a meeting, given $Ag1$'s intention to attend the meeting, and Ag_P's belief that the meeting is not related to an EU project. Rules in the goal base can also capture the sub-goal relationship. For example, if $Ag1$ intends to visit Leipzig for a meeting, then it derives the sub-goal goal of having funding for the visit. Finally, we assume argument construction from agents' bases is defined in some underlying logic.

Definition 7. *Let $\{Ag_1, \ldots, Ag_n\}$ be a set of agents, where for $i = 1 \ldots n$, Ag_i is equipped with a belief base $\mathcal{B}_i$, an intention base $\mathcal{I}_i$, and a goal base $\mathcal{G}_i$. For $i = 1$, let argument A be constructed from $\mathcal{B}i \cup \mathcal{G}i \cup \bigcup_{i=1}^{n} \mathcal{I}i$.*

If A is constructed only from $\mathcal{B}i$, then A is a belief argument *of Ag_i, otherwise A is a* goal argument *of Ag_i.*

In general, we write $bel(A)$ to denote the beliefs in A. We also write claim*(A) to identify an argument A's claim.*

The basic idea is that individual and system agents engage in argumentation-based conflict resolution (*persuasion*) dialogues to determine which amongst the arguments for beliefs and goals are justified in the *EAF*s of Section 2. The goals that are the claims of justified arguments are then adopted as intentions. In persuasion dialogues (reviewed in [3]) a proponent makes a claim — the *topic* of the dialogue — and (one or more) opponents attempt to persuade the proponent that the claim does not hold. In general such a dialogue d is a sequence of moves $m_1, \ldots, m_i, \ldots$, where the first move m_1 is a locution introducing the topic as an assertion or claim of an argument. Here, we simply assume that the topic of d can be referred to as *topic*(d). Dialogue *protocols* vary from model to model, and specify the legal moves at each stage of the dialogue, where a move can be an assertion of a proposition or an argument, a challenge to a premise in an argument, a concession of a proposition or argument, and so on. Models also vary on the rules for termination of a dialogue. However, in general, the arguments submitted and constructed (from the propositions asserted) during the course of a dialogue can be organised into an argumentation framework [15]. If an argument for the topic is justified, then the proponent wins the dialogue. Formalising dialogue models is to be addressed in future work. Here, we refer only to an *EAF* constructed on the basis of a dialogue.

Definition 8. *Let $d = m_1, \ldots, m_n$ be a terminated dialogue where* topic*(d) = α, and $AG = \{Ag_1, \ldots, Ag_m\}$ the participants in d. We say that the* EAF $\Delta = (Args, \mathcal{R}, \mathcal{D})$ *constructed on the basis of d, is:*

- *a belief EAF iff every argument in $Args$ is a belief argument of some $Ag \in AG$*
- *a goal EAF iff every argument in $Args$ is either a belief or goal argument of some $Ag \in AG$* [2]

3.2 Arguing about Beliefs and Goals

An agent's argument A for a desire may conflict with (and so mutually attack) an argument B for a normative goal. Arguments for punishment and reward goals may in turn attack A and so reinstate the argument B for the normative goal. The success of these attacks as defeats depends on argumentation over preferences between the arguments (corresponding to meta-level motivation-based argumentation over the relative utility of states in which the goals are realised).

Prior to agents submitting goal arguments in a dialogue, the beliefs in the argument justifying the goal may themselves by subject to debate [3]. In our running example, $Ag1$'s desire to stay on Waikiki beach is contingent on its belief that it will be hot in Hawaii. A system agent may successfully persuade $Ag1$ that it will be cool in Hawaii. Hence $Ag1$ will not submit the argument for its desire, precluding the possibility of norm violation (in Example 1 the outcome is in favour of $Ag1$'s argument that it *will* be hot). Furthermore, the beliefs in arguments for system goals may be challenged.

[2] Of course, in the limiting case where only arguments can be submitted as locutions, then each m_i in d corresponds to an argument in $Args$, and a protocol for d would require that m_i attack some m_j, $j < i$, or some attack between m_j and m_k, $j < i$, $k < i$

[3] Arguing over beliefs justifying a goal *prior* to arguing over the relative merits of goals precludes 'wishful thinking'; i.e., one wouldn't want that argumentation over which goals to adopt (which future state to realise) influences what is believed about the current state of the world.

Thus, an agent may successfully argue that the beliefs justifying a normative goal may be erroneous; hence the normative goal does not have to be adopted and unwarranted obstruction of the conflicting desire is prevented. Suppose arguments A and B for the conflicting goals of staying on Waikiki and staying at a cheap hotel have been submitted. Ag_P will not submit an argument C for the punishment goal of denying $Ag1$ funding for the Leipzig meeting, if $Ag1$ successfully persuades Ag_P that the meeting *is* related to an EU project. Again, this prevents unwarranted obstruction of $Ag1$'s intention to attend the meeting. Of course, Ag_P may then be motivated to submit an argument for an alternative punishment goal to enforce compliance.

Definition 9. *Let* $AG = \{Ag_1, \ldots, Ag_n\}$. *Then* A *is an* **agreed** *goal argument of* $Ag \in AG$ *if for every* $\alpha \in bel(A)$*:*
if there is a terminated dialogue d *with topic* α*, participants* $AG \subseteq \{Ag_1, \ldots, Ag_n\}$*, and* Δ *is a belief* EAF *constructed on the basis of* d*, then* α *is the conclusion of a justified argument of* Δ*.*

We now describe how argumentation over goals proceeds. Consider the case where a normative goal g' conflicts with a desire g (in the simplest case $g' \equiv \neg g$ in the underlying logic). In general, we say that the goal argument A' for g' *conflicts* with the goal argument A for g. In a goal *EAF*, A and A' attack each other since an agent can either adopt g and not g', or g' and not g.

Suppose such an EAF, where $Ag1$ submits A claiming 'stay on Waikiki beach', and A' claiming 'stay in cheap hotel', mutually attacks A. An enforcement agent can then submit an argument P for a punishment goal p, that, in the terminology of [11], *hinders* some intention of $Ag1$. In our running example, p = 'deny funding for meeting'. Now P does not directly attack on A's goal; it does so in the sense that if the attack succeeds, then $Ag1$ will not pursue its desire, and will comply with the norm. Note also, that the attack is a preference dependent asymmetric attack. $Ag1$ might argue (B) that it is of more value to him to stay on Waikiki beach then attend the meeting. That is, $B \twoheadrightarrow (P \rightharpoonup A)$, and it may now be that A *and* P are justified; $Ag1$ adopts its desire, *and* accepts the punishment. An alternative punishment may then need to be submitted to see if it has the required enforcing effect. Finally, an enforcement agent can submit an argument R for a reward goal r, that, in the terminology of [11], *benefits* some intention of Ag. For example r = 'provide the agent with a laptop', benefiting $Ag1$'s intention to have a laptop. R symmetrically attacks A. Either $Ag1$ accepts the reward and drops the desire, or vice versa.

Definition 10. *Let* $AG = \{Ag_1, \ldots, Ag_n\}$ *be a set of agents. Let* $Args_G = \bigcup_{i=1\ldots n}\{A|A$ *is a goal argument for* $Ag_i\}$. *Let* $\mathcal{I}_{AG} = \bigcup_{i=1\ldots n} \mathcal{I}i$. *Then:*

- conflicts $\subseteq Args_G \times Args_G$
- hinders $\subseteq Args_G \times \mathcal{I}_{AG}$
- benefits $\subseteq Args_G \times \mathcal{I}_{AG}$[4]

[4] Note that an agent's desires may 'internally' conflict. We so not here directly address conflict resolution in such cases. Note also that an agent's goals may benefit/hinder its own intentions.

Definition 11. *Let $AG = \{Ag_1, \ldots, Ag_n\}$, and for some $Ag \in AG$, let A be a goal argument of Ag, $\mathcal{I}$ the intention base of Ag.*
Let A' be the goal argument of some $Ag' \in AG$, $Ag' \neq Ag$. Then:

- A' goal attacks A *and* A goal attacks A' *if* conflicts*(A', A) or* benefits*(A', ι) for some $\iota \in \mathcal{I}$*
- A' goal attacks A *if* hinders*(A', ι) for some $\iota \in \mathcal{I}$*

We now specify some constraints on a dialogue that begins with a topic that is a goal proposed for adoption as an intention. We do so by expressing constraints on the goal *EAF* constructed on the basis of the dialogue. These are that the goal arguments are agreed, and can only be attacked by goal arguments as defined above, and only belief arguments are used in arguing over the relative merits of the goals.

Definition 12. *Let $AG = \{Ag_1, \ldots, Ag_n\}$ be a set of agents. Let d be a terminated dialogue with topic α, and participants $AG' \subseteq AG$, where:*

- *α is the conclusion of an agreed goal argument A of some agent $Ag \in AG'$.*
- *Δ = ($Args, \mathcal{R}, \mathcal{D}$) is the goal* EAF *constructed on the basis of d, where:*
  i) for any goal arguments $B, A \in Args$, (B, A) $\in \mathcal{R}$ iff A and B are agreed goal arguments, and B goal attacks A.
  ii) If (C,(B, A)) $\in \mathcal{D}$ then C is a belief argument for some agent in AG'

If the topic α of the dialogue is an agent's desire, and α is the claim of a justified argument in the dialogue's goal *EAF*, then α is updated to the agent's intention base, and any punishment goal that is the claim of a justified argument is updated to the corresponding enforcement agent's intention base. If α is not the claim of a justified argument, and there is a justified argument for a normative goal β, then β is updated to the agent's intention base, and any reward goal that is the claim of a justified argument is updated to the corresponding enforcement agent's intention base.

4 Instantiating the Framework

In this section we describe an example instantiation of the framework. Agent goals, beliefs and intentions are represented in [14]'s *argument based logic programming with defeasible priorities* (ALP-DP). An ALP-DP theory's arguments are defined as sequences of chained rules. Some rules can express priorities on other rules, so that one can construct *priority arguments* whose claims determine preferences between other mutually attacking arguments. Preferences between priority arguments can also be established on the basis of other priority arguments. [14] then defines the justified arguments of a theory under Dung's grounded semantics. In [12, 13] the arguments and attacks defined by an ALP-DP theory instantiate an *EAF*, and an equivalence result with the *EAF*'s justified arguments (under the grounded semantics) is shown. By giving an *EAF* semantics for ALP-DP one can, unlike [14], also:

1. characterise the justified arguments of an ALP-DP theory under the preferred semantics; and

2. model preference dependent asymmetric attacks.

Both these features are employed when instantiating an *EAF*. Note that ALP-DP models both negation as failure and strict negation. To simplify the presentation, we describe a restricted version of ALP-DP — ALP-DP* — which does not include negation as failure.

Definition 13. *Let (S, D) be a ALP-DP* theory where S is a set of strict rules of the form $s : L_0 \wedge \ldots \wedge L_m \rightarrow L_n$, D a set of defeasible rules $r : L_0 \wedge \ldots \wedge L_j \Rightarrow L_n$, and:*

- *Each rule name r (s) is a first order term. Henceforth,* head*(r) denotes the consequent L_n of the rule named r.*
- *Each L_i is an atomic first order formula, or such a formula preceded by strong negation $\neg$.*

Strict rules represent information that is beyond debate (note that neither $\rightarrow$ nor $\Rightarrow$ admit contraposition). We also assume that the language contains a two-place predicate symbol $\prec$ for expressing priorities on rule names, and that any S includes the following strict rules expressing the properties of a strict partial order on $\prec$:

- $o1 : (x \prec y) \wedge (y \prec z) \rightarrow (x \prec z)$
- $o2 : (x \prec y) \wedge \neg(x \prec z) \rightarrow \neg(y \prec z)$
- $o3 : (y \prec z) \wedge \neg(x \prec z) \rightarrow \neg(x \prec y)$
- $o4 : (x \prec y) \rightarrow \neg(y \prec x)$

Definition 14. *An argument A based on the theory (S, D) is:*

1. *a finite sequence $[r_0, \ldots, r_n]$ of ground instances of rules such that*
   - *for every i ($0 \leq i \leq n$), for every literal L_j in the antecedent of r_i there is a $k < i$ such that* head*(r_k) = L_j.*
     We say that claim*(A) =* head*(r_n), and if* head*(r_n) = $x \prec y$ then A is called a 'singleton priority argument'.*
   - *no distinct rules in the sequence have the same head*

   or
2. *a finite sequence $[r_{0_1}, \ldots r_{n_1}, \ldots, r_{0_m}, \ldots r_{n_m}]$, such that for $i = 1 \ldots m$, $[r_{0_i}, \ldots r_{n_i}]$ is a singleton priority argument. We say that A is a 'composite priority argument' and* claim*(A) =* head*(r_{n_1})...*head*(r_{n_m}) is the ordering claimed by A*

In [14], arguments are exclusively defined by item 1. We additionally define composite priority arguments so that an ordering, and hence a preference, can be claimed by a single argument rather than a set of arguments (as in [14]).

Definition 15. *For any arguments A, A' and literal L:*

- *A is strict iff it does not contain any defeasible rule; it is defeasible otherwise.*
- *L is a conclusion of A iff L is the head of some rule in A*
- *If T is a sequence of rules, then $A + T$ is the concatenation of A and T*

Note that an argument has only one claim, but may have many conclusions corresponding to the heads of the contained rules. We now instantiate the abstract definition 7 of an agent and its constructed arguments. Note that intentions are represented by the goal arguments that have previously been used to justify their adoption. Hence, an agent's goal arguments will be constructed from its belief and goal base, and the claims (named by the name of the rule whose head is the claim) of intention arguments in all agents' intention bases.

Definition 16. *Let $\{Ag_1, \ldots, Ag_n\}$ be a set of agents, where for $i = 1 \ldots n$:*
- $\mathcal{B}_i$ and $\mathcal{G}_i$ are ALP-DP theories, and $\mathcal{I}_i$ is a set of arguments.*
- A is a belief argument of Ag_i iff it is based on $\mathcal{B}_i$
- A is a goal argument of Ag_i iff it is based on $\mathcal{B}_i \cup \mathcal{G}_i \cup \bigcup_{i=1 \ldots n}\{r : claim(B) | B \in \mathcal{I}_i, head(r) = claim(B)\}$

[14] motivates definition of attacks between arguments that account for the ways in which arguments can be extended with strict rules:

Definition 17. *$A1$ attacks $A2$ on the pair $(L, \neg L)$ if there are sequences S1 and S2 of strict rules such that $A1 + S1$ is an argument with conclusion L and $A2 + S2$ is an argument with a conclusion $\neg L$.*

In the following example illustrating attacks between belief arguments, we will without loss of generality simply assume that all beliefs are contained in a single theory. Only in the example at the end of this section, in which we illustrate argumentation over goals, will we identify the individual agents involved. Following [14], every rule with terms $t_1, \ldots, t_n$ is named with a function expression $r(t_1, \ldots, t_n)$ where r is the rules's informal name. For example, $r(p(X, Y), q(X, Y))$ names the rule $p(X, Y) \Rightarrow q(X, Y)$. To maintain readability we will only write the function-symbol part of the rule name, and as an abuse of notation, arguments will be represented as sequences of rule names rather than the rules these names identify.

Example 2. Let $tr(X, Y)$, $st(X, Y)$ and $ra(X, Y)$ respectively denote that X is more trustworthy, statistically accurate, and rational than Y.
Let $S = \{o1 \ldots o4\} \cup \{s1 : temp(X, cool) \rightarrow \neg temp(X, hot),$
$s2 : temp(X, hot) \rightarrow \neg temp(X, cool)\}$.
Let $D = \{bbc :\Rightarrow temp(hawaii, cool),$
$cnn :\Rightarrow temp(hawaii, hot),$
$c1 :\Rightarrow tr(bbc, cnn),$
$d1 :\Rightarrow st(cnn, bbc),$
$c2 : tr(X, Y) \Rightarrow Y \prec X,$
$d2 : st(X, Y) \Rightarrow Y \prec X,$
$e1 :\Rightarrow ra(d2, c2),$
$e2 : ra(X, Y) \Rightarrow Y \prec X\}$
$A = [bbc]$, $B = [cnn]$, $C = [c1, c2]$, $D = [d1, d2]$.
$E = [e1, e2]$ with conclusions $ra(d2, c2)$ and $c2 \prec d2$, and claim $c2 \prec d2$.

A and B attack each other since $A + s1$ has conclusion $\neg temp(hawaii, hot)$ and B has conclusion $temp(hawaii, hot)$. C and D attack each other since C has conclusion $cnn \prec bbc$ and $D + o4$ has conclusion $\neg(cnn \prec bbc)$

We now define the relations *conflicts*, *hinders* and *benefits*, and goal attacks for ALP-DP* goal arguments. Note that the notion of *benefits* requires that a goal argument of a rewarding agent be extended (as in the definition of attack) with strict rules that link the reward goal to the intention that it benefits (this will be illustrated in the example concluding this section).

Definition 18. *Let A be a goal argument of an agent Ag and $\mathcal{I}$ the intention base of Ag.*
Let B be any goal argument of an agent Ag', where $\mathcal{B}'$ = (S', D') is the belief base of Ag'. We say that:
• conflicts*(B, A) if B attacks A as in definition 17.*
For any $I \in \mathcal{I}$:
• hinders*(B, I) if B attacks I as in definition 17*
• benefits*(B, I) if* claim*$(B + S1)$* = claim*(I) for some possibly empty sequence of strict rules $S1$ in S'*
Then:

- *B and A goal attack each other on the pair $(claim(B), claim(A))$ if* conflicts*(B, A)*
- *B and A goal attack each other on the pair $(claim(B), claim(A))$ if* benefits*(B, I) for some $I \in \mathcal{I}$*
- *B goal attacks A on the pair $(claim(B), claim(A))$ if* hinders*(B, I) for some $I \in \mathcal{I}$*

To determine a preference amongst attacking arguments, [14] defines the sets of relevant *defeasible* rules to be compared, and an ordering on these sets. Here, the ordering on such sets is based on the ordering claimed by a given priority argument.

Definition 19. *If $A + S$ is an argument with conclusion L, the defeasible rules $R_L(A + S)$ **relevant** to L are:*

1. *$\{r_d\}$ iff A includes defeasible rule r_d with head L*
2. *$R_{L_1}(A + S) \cup \ldots \cup R_{L_n}(A + S)$ iff A is defeasible and S includes a strict rule s : $L_1 \wedge \ldots \wedge L_n \rightarrow L$*

Definition 20. *Let C be a priority argument claiming the ordering $\prec$. Let R and R' be sets of defeasible rules. Then $R' > R$ iff $\forall r' \in R'$, $\exists r \in R$ such that $r \prec r'$.*

Intuitively, R can be made better by replacing some rule in R with any rule in R', while the reverse is impossible. Now, given two arguments A and B, it may be that for belief arguments they attack on more than one conclusion. For goal arguments they goal attack on a single pair of conclusions (the goals claimed by the arguments). Given a priority ordering $\prec$ claimed by argument C, we say that A is preferred$_{\prec}$ to B if for every pair (L, L') of conclusions on which they attack, the set of A's defeasible rules relevant to L is stronger ($>$) than the set of B's defeasible rules relevant to L'.

Definition 21. *Let C be a priority argument claiming $\prec$. Let $(L_1, L'_1), \ldots, (L_n, L'_n)$ be the pairs on which A attacks, or goal attacks B, where for $i = 1 \ldots n$, L_i and L'_i are conclusions in A and B respectively. Then A is preferred$_\prec$ to B if for $i = 1 \ldots n$, $R_{L_i}(A + S_i) > R_{L'_i}(B + S'_i)$*

In example 2, C and D attack each other on the pair ($cnn \prec bbc$, $\neg(cnn \prec bbc)$), and $R_{cnn \prec bbc}(C)$ = $\{c2\}$, $R_{\neg(cnn \prec bbc)}(D)$ = $\{d2\}$. E claims $c2 \prec d2$, and so D is preferred$_{c2 \prec d2}$ to C. Note also, that given C, A is preferred$_{cnn \prec bbc}$ to B, and given D, B is preferred$_{bbc \prec cnn}$ to A. We can now instantiate an *EAF* with the arguments, their attacks, and priority arguments claiming preferences and so attacking attacks:

Definition 22. *The* EAF *($Args$, $\mathcal{R}$, $\mathcal{D}$) for a theory (S, D) is defined as follows. $Args$ is the set of arguments given by definition 14, and $\forall A, B, C \in Args$:*

1. *(C,(B, A)) $\in \mathcal{D}$ iff C claims $\prec$ and A is preferred$_\prec$ to B*
2. *(A, B),(B, A) $\in \mathcal{R}$ if A and B attack as in definition 17, or A and B goal attack as in definition 18*

The belief *EAF* obtained by the arguments and attacks for our running example is shown in figure 1a). $\{E, D, B\}$ is the single preferred extension of the *EAF*. We can now constrain a goal EAF constructed on the basis of a dialogue between agents, as defined in definition 12.

5 An Extended Example

We now illustrate the previous section's formalism with an extended version of our Hawaiian example, in which we assume that every goal argument is agreed.

In what follows we use the following shorthand:

ha = 'Hawaii', wa = 'Waikiki beach', le = 'Leipzig', att, = 'attend', $conf$ = 'conference', $meet$, = 'meeting', $cheap$ = 'cheap hotel', lap = 'laptop', $fund$ = 'have funding', and $deny\_f$ = 'deny funding'.

Also, predicates may refer to the agents themselves. For example, $att(ag, conf, ha)$ denotes the goal of ag to attend a conference in Hawaii. Also, variables will begin with uppercase letters and constants with lowercase letters. For example, $deny\_f(ag_P, AgX, meet, L)$ denotes the goal of agent ag_P to deny funding for any agent AgX to attend a meeting in some location L.

Let $\{ag_a, ag_N, ag_P, ag_R\}$ be a set of agents. We describe each agent's knowledge bases. Note that we may not show all the goal rules used to construct arguments in the intention base of each agent. Also, as before, we may simply write the rule name rather than the rule the name identifies.

ag_a:
$\mathcal{I}$ =
{ [$ia1 :\Rightarrow att(ag_a, conf, ha)$], [$ia2 :\Rightarrow att(ag_a, meet, le)$],
[$ia2 :\Rightarrow att(ag_a, meet, le), ia3 : att(ag_a, meet, le) \Rightarrow funds(ag_a, meet, le)$],
[$ia4 :\Rightarrow have(ag_a, lap)$] }

$\mathcal{G}$ =
{ $ga1 : temp(ha, hot) \wedge att(ag_a, conf, ha) \Rightarrow stay(ag_a, wa)$}

$\mathcal{B}$ =
{ $ba0 :\Rightarrow temp(ha, hot)$,
$ba1 :\rightarrow norm\_des(gn1, ga1)$,
$ba\_self : norm\_des(X, Y) \Rightarrow X \prec Y$,
$ba2 :\Rightarrow project\_funds(high)$,
$ba3 : project\_funds(high) \Rightarrow except(ba\_self, bn\_social)$,
$ba\_excep : except(X, Y) \Rightarrow Y \prec X$,
$ba4 :\Rightarrow gp1 \prec ga1$}

ag_N:
$\mathcal{I} = \emptyset$
$\mathcal{G}$ = { $gn1 : att(AgX, conf, L) \Rightarrow stay(AgX, cheap, L)$}
$\mathcal{B}$ = { $bn1 : stay(AgX, cheap, ha) \rightarrow \neg stay(AgX, wa)$,
$bn2 :\rightarrow norm\_des(gn1, ga1)$,
$bn\_social : norm\_des(X, Y) \Rightarrow Y \prec X$ }

ag_P:
$\mathcal{I} = \emptyset$
$\mathcal{G}$ = { $gp1 : att(AgX, meet, L) \wedge \neg type(meet, eu, L)$
$\Rightarrow deny\_f(ag_P, AgX, meet, L)$}
$\mathcal{B}$ = { $bp1 :\Rightarrow \neg type(meet, eu, le)$,
$bp2 : deny\_f(ag_P, AgX, meet, L) \rightarrow \neg funds(AgX, meet, L)$}

ag_R:
$\mathcal{I} = \emptyset$
$\mathcal{G}$ = { $gr1 : have(AgX, lap) \Rightarrow provide(ag_R, AgX, lap)$}
$\mathcal{B}$ = { $br1 : provide(ag_R, AgX, lap) \rightarrow have(AgX, lap)$,
$br2 :\rightarrow rew\_des(gr1, ga1)$,
$br\_rew\_suffice : rew\_des(X, Y) \Rightarrow Y \prec X$ }

1) ag_a initiates a dialogue with goal argument $A1$ = [$ba0, ia1, ga1$] claiming the goal $stay(ag_a, wa)$, having already persuaded a system agent that it will indeed be hot in Hawaii.

2) ag_N submits $A2$ = [$ia1$,$gn1$] ($AgX = ag_a$, $L = ha$), where $A2$ and $A1$ goal attack each other (see figure 2) on the pair
($stay(ag_a, cheap, ha), stay(ag_a, wa)$).
This symmetric goal attack is based on *conflicts* ($A2, A1$) which obtains because $A2 +$ [$bn1$] and $A1$ attack (as in def.17) on the conclusion pair ($\neg stay(ag_a, wa), stay(ag_a, wa)$)
ag_N also submits the social ordering argument $B1$ = [$bn2, bn\_social$] claiming $ga1 \prec gn1$, and so $B1 \twoheadrightarrow (A1 \rightharpoonup A2)$.

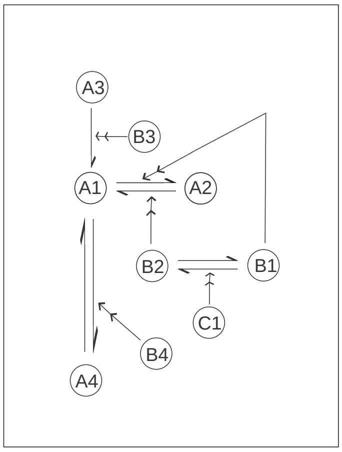


Fig. 2. *EAF* based on argumentation based dialogue over goals

3) ag_a submits:

- the selfish ordering argument $B2$ = $[ba1, ba\_self]$ claiming $gn1 \prec ga1$, and so $B2 \twoheadrightarrow (A2 \rightharpoonup A1)$

- an argument claiming that the selfish behaviour type is preferred to the social behaviour type given the exceptional circumstances in which the remaining project budget is high:

$C1$ = $[ba2, ba3, ba\_excep,]$ claiming $bn\_social \prec ba\_self$, and so $C1 \twoheadrightarrow (B1 \rightharpoonup B2)$.

The single preferred extension contains $A1$

4) ag_P attempts to enforce compliance by submitting $A3$ = $[ia2, bp1, gp1]$ given that it is agreed that the meeting is not an Eu project meeting.

$A3 + [bp2]$ attacks (as in def.17), and so hinders, ag_a's intention $[ia2, ia3]$. Hence, $A3$ goal attacks $A1$ on the pair $(deny\_f(ag_P, ag_a, meet, le), stay(ag_a, wa))$.

5) However, ag_a prefers to stay on the beach and be denied funding by ag_P for the leipzig meeting. It may be that ag_a has another source of funding in mind. We do not encode the rationale for the preference, but simply assume the priority argument $B3$ = $[ba4]$ claiming $gp1 \prec ga1$. Hence $B3 \twoheadrightarrow (A3 \rightharpoonup A1)$. Since $A3$'s attack on $A1$ is asymmetric:

The single preferred extension contains $A1$ **and** $A3$

6) ag_R attempts to enforce compliance with $A4$ = $[ia4, gr1]$ offering to provide ag_a with a laptop. This benefits ag_a's intention to have a laptop since *claim*($[ia4, gr1]$ + $[br1]$) = *claim*$[ia4]$. Hence, $A4$ and $A1$ goal attack each other ($A4 \rightleftharpoons A1$) on the pair $(provide(ag_R, ag_a, lap), stay(ag_a, wa))$.

ag_R believes the reward is of sufficient strength that ag_a will prefer the reward to staying on Waikiki beach. ag_R submits $B4$ = $[br2, br\_rew\_suffice]$ claiming $ga1 \prec gr1$. Hence, $B4 \twoheadrightarrow (A1 \rightharpoonup A4)$. This is accepted by ag_a and the dialogue terminates.

The single preferred extension contains $A2$ and $A4$
ag_a's intention set can then be updated with $A2$. ag_R's intention set can then be updated with $A4$. ag_a intends now to book a cheap room in Hawaii, and ag_R intends to provide ag_a with a laptop.

6 Conclusions

In this paper we have proposed a framework for argumentation-based resolution of conflicts in normative multi-agent systems, and have illustrated instantiation of the framework with a logic programming formalism. The framework provides for agents to argue over the beliefs justifying goals, conflicting preferences brought to bear in argumentation over beliefs, and metalevel motivational argumentation over the states represented by desire based goals, and normative, punishment and reward goals argued for by other agents. In this way, unwarranted obstruction of individual agents' desires is precluded, and enforcement of compliance can appropriately account for the motivations of the agents and erroneously held beliefs about the contexts in which the agents find themselves.

As mentioned in Section 1, existing approaches to argumentation-based resolution of conflicts amongst goals ([2],[10],[16]) do not model social mechanisms deployed to enforce compliance with norms. In [16], norms are represented as bridge rules that describe the relationships between mental attitudes. Argumentation based resolution of conflicts amongst goals derived using these rules exploits a preference relation on these rules. In [2], only conflicts amongst desire based goals are addressed. Argumentation over the beliefs that justify desires conforms to the Dung semantics. However selection of desires does not account for their relative importance and does not conform to the Dung semantics. Rather, the maximal (under set inclusion) sets of desires that can be consistently realised are chosen. However, goal selection *does* account for the feasibility of plans for realising goals, and this is a factor that our work needs to account for in future work.

Future work will also investigate instantiation of the framework by formalisms with explicit BDI type modalities. Further work is also required before evaluation of the framework based on prototypical implementations can proceed. In particular, we intend development of argument game proof theories, algorithms and dialogue protocols for *EAF*s. Since *EAF*s inherit the fundamental results shown for Dung frameworks, our approach will adopt the methodologies deployed in specification of game based algorithms [18] and protocols [15] based on the Dung semantics. We also believe that our approach is applicable to resolution of conflicts arising between an individual agent's conflicting desires, and between conflicting norms. Both cases often require reasoning about abstract values and motivations. Furthermore, conflict resolution may lead to refinement and evolution of a system's norms. Finally, one of the key novel features of our framework is that an agent's decision as to which goals to pursue is influenced by other agents' goals. We believe that we can abstract from the normative application of the framework to consider other contexts in which the impact of other agents' goals can be modelled through argumentation based mechanisms.

Acknowledgements The research described in this paper is partly supported by the European Commission Framework 6 funded project CONTRACT (INFSO-IST-034418). The opinions expressed herein are those of the named authors only and should not be taken as necessarily representative of the opinion of the European Commission or CONTRACT project partners.

References

1. L. Amgoud. Using Preferences to Select Acceptable Arguments. In: *Proc. 13th European Conference on Artificial Intelligence*, 43-44, 1998.
2. L. Amgoud and S. Kaci. On the Generation of Bipolar Goals in Argumentation-Based Negotiation. In: *Proc. 1st Int. Workshop on Argumentation in Multi-Agent Systems*, 2004.
3. ASPIC Deliverable D2.1: Theoretical frameworks for argumentation. http://www.argumentation.org/ Public Deliverables.htm, June 2004.
4. T. J. M. Bench-Capon. Persuasion in Practical Argument Using Value-based Argumentation Frameworks, *Journal of Logic and Computation*, 13(3), 429-448, 2003.
5. A. Bondarenko, P. M. Dung, R. A. Kowalski and F. Toni. An abstract, argumentation-theoretic approach to default reasoning, *Artificial Intelligence*, 93:63-101, 1997.
6. J. Broersen, M. Dastani, J. Hulstijn and L. W. N. van der Torre. Goal Generation in the BOID Architecture. In: Cognitive Science Quarterly Journal, 2(3-4), 428-447, 2002.
7. M. Dastani and L. van der Torre. Programming BOID-Plan Agents: Deliberating about Conflicts among Defeasible Mental Attitudes and Plans. In: *Proc 3rd Int. Joint Conference on Autonomous Agents and Multiagent Systems*, 706-713, 2004.
8. P. M. Dung. On the acceptability of arguments and its fundamental role in nonmonotonic reasoning, logic programming and n-person games, *Artificial Intelligence*,77:321-357, 1995.
9. M. d'Inverno and M. Luck. Understanding agent systems. *2nd edn Springer-Verlag*.
10. A. Kakas and P. Moraitis. Argumentation based decision making for autonomous agents. In: *Proc. Second international joint conference on autonomous agents and multiagent systems*, 883–890, 2003.
11. F. Lopez Y Lopez, M. Luck and M. D'Inverno. A normative framework for agent-based systems. In: *J. Computational and Mathematical Organization Theory*, 12 (2-3):227–250, 2006.
12. S. Modgil. An Abstract Theory of Argumentation That Accommodates Defeasible Reasoning About Preferences. In: *9th European Conference on Symbolic and Quantitative Approaches to Reasoning with Uncertainty*, 648–659, 2007.
13. S. Modgil. *Reasoning About Preferences in Argumentation Frameworks*. Technical Report: http://www.dcs.kcl.ac.uk/staff/modgilsa/ArguingAboutPreferences.pdf
14. H. Prakken and G. Sartor. Argument-based extended logic programming with defeasible priorities, *Journal of Applied Non-Classical Logics*,7:25–75, 1997.
15. H. Prakken. Coherence and flexibility in dialogue games for argumentation. In: *Journal of logic and computation* 15 (6):1009–1040 , 2005.
16. D. Gaertner, F. Toni. Conflict-free normative agents using assumption-basedargumentation. In: *Proc. 4th International Workshop on Argumentation in Multi-Agent Systems*, Hawaii, 2007.
17. Y. Moses and M. Tennenholtz. Artificial Social Systems. In: *Computers and AI*, 14(6), 533–562, 1995.
18. G. Vreeswijk. An algorithm to compute minimally grounded and admissible defence sets in argument systems. In: *Proc. 1st International Conference on Computational Models of Argument*,109-120, 2006.

Contract Formation through Preemptive Normative Conflict Resolution[⋆]

Wamberto W. Vasconcelos[†] and Timothy J. Norman[‡]

Dept. of Computing Science, University of Aberdeen, AB24 3UE, United Kingdom
[†]w.w.vasconcelos@abdn.ac.uk, [‡]t.j.norman@abdn.ac.uk

Abstract. We explore a rule-based formalisation for contracts: the rules capture conditional norms, that is, they describe situations arising during the enactment of a multi-agent system, and norms that arise from these situations. However, such rules may establish conflicting norms, that is, norms which simultaneously prohibit and oblige (or prohibit and permit) agents to perform particular actions. We propose to use a mechanism to detect and resolve normative conflicts in a preemptive fashion: these mechanisms are used to analyse a contract and suggest "amendments" to the clauses of the contract. These amendments narrow down the scope of influence of norms and avoid normative conflicts. Agents propose rules and their amendments, leading to a contract in which no conflicts may arise.

1 Introduction

We explore a rule-based formalisation for contracts: the rules capture conditional norms, that is, they describe situations arising during the enactment of a multi-agent system (MAS), and norms that arise from these situations. However, such rules may establish conflicting norms, that is, norms which simultaneously prohibit and oblige (or prohibit and permit) agents to perform particular actions. We propose to use a mechanism to detect and resolve normative conflicts in a preemptive fashion: these mechanisms are used to analyse a contract and suggest "amendments" to the clauses of the contract. These amendments narrow down the scope of influence of norms and avoid normative conflicts.

We envisage a scenario in which agents propose rules which will make up a contract. Agents, however, may already be committed to existing contracts when they are negotiating the terms of a new contract. Furthermore, these agents may not want to divulge the terms of the contracts they have established, that is, they may not want to justify why they need to propose amendments to a contract.

[⋆] This research was sponsored by the U.S. Army Research Laboratory and the U.K. Ministry of Defence and was accomplished under Agreement Number W911NF-06-3-0001. The views and conclusions contained in this document are those of the author(s) and should not be interpreted as representing the official policies, either expressed or implied, of the U.S. Army Research Laboratory, the U.S. Government, the U.K. Ministry of Defence or the U.K. Government. The U.S. and U.K. Governments are authorised to reproduce and distribute reprints for Government purposes notwithstanding any copyright notation hereon.

Dagstuhl Seminar Proceedings 09121
Normative Multi-Agent Systems
http://drops.dagstuhl.de/opus/volltexte/2009/1920

The structure of this paper is as follows. In Section 2 we introduce norm-governed multi-agent systems, also presenting our account of norms and their formal underpinnings. Section 3 formally presents the syntax and semantics of contracts as a set of rules; additionally that section provides a computational model for contract enactments. In Section 4 we present mechanisms to detect and resolve normative conflicts, using unification and constraint satisfaction techniques. In Section 5 we introduce our preemptive approach to contract formation, whereby agents exchange messages with contract clauses and amendments to these. We compare our approach with related work in Section 6 and conclude in Section 7, where we also give directions for future work.

2 Norm-Governed Multi-Agent Systems

The design of complex multi-agent systems is greatly facilitated if we move away from individual components and, instead, regard them as belonging to stereotypical classes or categories of components. One way to carry out this classification/categorisation is through the use of *roles* as introduced in, *e.g.*, [4, 17] – an agent takes on a role within a society or an organisation, and this role defines a pattern of behaviour to which any agent ought to conform. For instance, within a humanitarian relief force, there are roles such as medical assistant, member of mine clearance team, and so on, and agents adopt these roles (possibly more than one) as they join the force. When agents adopt roles they commit themselves to the roles' expected behaviours, with associated sanctions and rewards. We shall make use of two finite, non-empty sets, $\mathit{Agents} = \{a_1, \ldots, a_n\}$ and $\mathit{Roles} = \{r_1, \ldots, r_m\}$, representing, respectively, the sets of agent identifiers and role labels.

The building blocks of our formalism are *terms*:

Definition 1. *A term, denoted as τ, is any variable x, y, z (with or without subscripts) or any construct $f^n(\tau_1, \ldots, \tau_n)$, where f^n is an n-ary function symbol and $\tau_1, \ldots, \tau_n$ are terms.*

Terms f^0 stand for *constants* and will be denoted as a, b, c (with or without subscripts). We shall also make use of numbers and arithmetic functions to build our terms; arithmetic functions may appear infix, following their usual conventions. We adopt Prolog's convention [1] using strings starting with a capital letter to represent variables and strings starting with a small letter to represent constants. Some examples of terms are Price (a variable) and $\mathit{send}(a, B, \mathit{inform}(c))$ (a function).

We also define *atomic formulae*:

Definition 2. *An atomic formula, denoted as φ, is any construct $p^n(\tau_1, \ldots, \tau_n)$, where p^n is an n-ary predicate symbol and $\tau_1, \ldots, \tau_n$ are terms.*

When the context makes it clear what n is we can drop it. p^0 stands for propositions. We shall employ arithmetic relations (*e.g.*, $=$, $\neq$, and so on) as predicate symbols, and these will appear in their usual infix notation. We also make use of atomic formulae built with arithmetic relations to represent *constraints* on variables – these atomic formulae have a special status, as we explain below. We give a definition of our constraints, a subset of atomic formulae:

Definition 3. *A constraint γ is an infix binary atomic formula $\tau \lhd \tau'$, where $\lhd$ is any of the symbols $=, \neq, >, \geq, <$, or $\leq$.*

We shall denote a possibly empty set of constraints as $\Gamma = \{\gamma_0, \ldots, \gamma_n\}$ and it stands for a *conjunction* of the constraints, that is, $\bigwedge_{i=0}^{n} \gamma_i$. Some sample constraints are $X < 120$ and $X < (Y + Z)$. To improve readability, constraints of the form $\{10 \leq X, X \leq 45\}$ will be written as $\{10 \leq X \leq 45\}$.

We need an account of those actions performed by agents:

Definition 4. *An action tuple is $\langle a\!:\!r, \bar{\varphi}\rangle$ where*

- *$\bar{\varphi}$, a ground first-order atomic formula, representing an action*
- *$a \in$ Agents is the agent who did $\bar{\varphi}$*
- *$r \in$ Roles is the role played by the agent a when it did $\bar{\varphi}$*

Agents perform their actions in a distributed fashion, contributing to the overall enactment of the MAS. However, for ease of presentation, we make use of a global (centralised) account for all actions taking place; therefore, it is important to record the authorship of actions.

2.1 A Representation for Norms

In this section we introduce our representation of norms. We extend our previous work [22, 23], adopting the notation of [17] for specifying norms, complementing it with *constraints* [9]. Constraints are used to further *refine* the scope of influence of norms on actions.

We associate constraints with first-order formulae, imposing restrictions on their variables. We represent this association as $\varphi \circ \Gamma$, as in, for instance, $deploy(s_1, X, Y) \circ \{10 \leq X \leq 50, 5 \leq Y \leq 45\}$. When Γ is empty, we will simply drop it from our formulae. Norms are thus defined:

Definition 5. *A norm ω is any construct*

- *$\mathsf{O}_{\alpha:\rho}\varphi \circ \Gamma$ (an obligation),*
- *$\mathsf{P}_{\alpha:\rho}\varphi \circ \Gamma$ (a permission), or*
- *$\mathsf{F}_{\alpha:\rho}\varphi \circ \Gamma$ (a prohibition),*

where α, ρ are terms, φ is a first-order atomic formula and Γ is a possibly empty set of constraints.

Term α identifies the agent(s) to whom the norm is applicable and ρ is the role of such agent(s). $\mathsf{O}_{\alpha:\rho}\varphi \circ \{\gamma_0, \ldots, \gamma_n\}$ thus represents an obligation on agent α taking up role ρ to bring about φ, subject to *all* constraints γ_i, $0 \leq i \leq n$. The γ_i terms express constraints on variables of φ.

For simplicity, in our discussion we assume an implicit universal quantification over variables in ω. For instance, $\mathsf{P}_{A:R}deploy(X, b, c)$ stands for $\forall A \in$

$Agents.\forall R \in Roles.\forall X.\mathsf{P}_{A:R} deploy(X, b, c)$. However, our proposal can be naturally extended to cope with arbitrary quantifications. Obligations normally require the arguments of their actions to be existentially quantified, as in, for instance

$$\forall A \in Agents.\forall R \in Roles.\exists X.\exists Y.\exists Z.\mathsf{O}_{A:R} deploy(X, Y, Z)$$

Quantifications on agent ids and role labels may be universal or existential, and the relative ordering of quantifications defines the applicability of the norm, following the usual first-order logic semantics [5, 15].

We propose to formally represent from a global perspective the normative positions [19] of all agents taking part in a virtual society. By "normative position" we mean the "social burden" associated with individuals [6], that is, their obligations, permissions and prohibitions.

2.2 Substitutions, Unification and Constraint Satisfaction

We use first-order unification [5] and constraint satisfaction [10] as the building blocks of our mechanisms. Unification allows us *(i)* to detect whether norms are in conflict and *(ii)* to detect the set of actions that are under the influence of a norm. Initially, we define substitutions:

Definition 6. *A substitution σ is a finite and possibly empty set of pairs x/τ, where x is a variable and τ is a term.*

We define the application of a substitution in accordance with [5]. In addition, we describe how substitutions are applied to sets of constraints and norms (X stands for O, P or F):

1. $c \cdot \sigma = c$ for a constant c.
2. $x \cdot \sigma = \tau \cdot \sigma$ if $x/\tau \in \sigma$; otherwise $x \cdot \sigma = x$.
3. $p^n(\tau_0, \ldots, \tau_n) \cdot \sigma = p^n(\tau_0 \cdot \sigma, \ldots, \tau_n \cdot \sigma)$.
4. $\{\gamma_0, \ldots, \gamma_n\} \cdot \sigma = \{\gamma_0 \cdot \sigma, \ldots, \gamma_n \cdot \sigma\}$
5. $(\mathsf{X}_{\alpha:\rho}\varphi \circ \Gamma) \cdot \sigma = (\mathsf{X}_{(\alpha\cdot\sigma):(\rho\cdot\sigma)}(\varphi \cdot \sigma) \circ (\Gamma \cdot \sigma))$.

A substitution σ is a *unifier* of two terms τ_1, τ_2, if $\tau_1 \cdot \sigma = \tau_2 \cdot \sigma$. Unification is a fundamental problem in automated theorem proving and many algorithms have been proposed [5]; recent work offers means to obtain unifiers efficiently. We use unification in the following way:

Definition 7. *$unify(\tau_1, \tau_2, \sigma)$ holds iff $\tau_1 \cdot \sigma = \tau_2 \cdot \sigma$, for some σ. $unify(p^n(\tau_0, \ldots, \tau_n), p^n(\tau'_0, \ldots, \tau'_n), \sigma)$ holds iff $unify(\tau_i, \tau'_i, \sigma), 0 \leq i \leq n$.*

The *unify* relationship checks if a substitution σ is indeed a unifier for τ_1, τ_2, but it can also be used to find σ. We assume that *unify* is a suitable implementation of a unification algorithm which *(i)* always terminates (possibly failing, if a unifier cannot be found); *(ii)* is correct; and *(iii)* has a linear computational complexity.

We make use of existing constraint satisfaction techniques [9, 10] to implement a *satisfy* predicate which checks if a given set of constraints admits one solution, that is, the predicate holds if the variables of the constraints admit at least one value which simultaneously fulfills all constraints:

Definition 8. *$satisfy(\{\gamma_0, \ldots, \gamma_n\})$ holds iff $\bigwedge_{i=0}^{n}(\gamma_i \cdot \sigma)$ is true for some σ.*

This predicate can be implemented via different "off-the-shelf" constraint satisfaction libraries; for instance, it can be defined via the built-in `call_residue_vars`/2 predicate, available in SICStus Prolog [21] as:

$$satisfy(\{\gamma_0, \ldots, \gamma_n\}) \leftarrow \texttt{call\_residue\_vars}((\gamma_0, \ldots, \gamma_n), \_)$$

Predicate `call_residue_vars`(*Goals*, *Vars*) evaluates if *Goals* admit one possible solution, collecting in *Vars* the list of residual variables that have blocked goals or attributes attached to them. In our definition above, the value of *Vars* is not relevant, as we simply want to know if *Goals* are satisfiable.

2.3 Meaning of Norms

We explain the meaning of our norms in terms of their relationships with action tuples of global enactment states. We define when an individual action tuple is within the scope of influence of a norm – we do so via the logic program of Fig. 1. It defines predicate *inScope* which holds if its first argument, an action tuple (in the format of Def. 4), is within the influence of a norm ω (in the format of Def. 5), its second parameter. Lines 2 and 3 define, respectively, the format of *Action* and ω (where X is either P, F or O). Line 4 tests *(i)* if the agent performing the action and its role unify with α, ρ of ω and *(ii)* if the actions $\bar{\varphi}$ and φ unify. Line 5 checks if the constraints on ω (instantiated with the substitution σ obtained in line 4) can be satisfied.

```
1 $inScope(Action, \omega) \leftarrow$
2     $Action = \langle a : r, \bar{\varphi} \rangle \wedge$
3     $\omega = \mathsf{X}_{\alpha:\rho}\varphi \circ \Gamma \wedge$
4     $unify(\langle a, r, \bar{\varphi} \rangle, \langle \alpha, \rho, \varphi \rangle, \sigma) \wedge$
5     $satisfy(\Gamma \cdot \sigma)$
```

Fig. 1. Check if Action is within Influence of a Norm

Agents may experience difficulties if an action is simultaneously within the scope of influence of a prohibition and an obligation (or a prohibition and a permission). In such circumstances, whatever the agents do or refrain from doing, may give rise to an enactment state that is not norm-compliant. The agents will thus violate a norm, and will be subject to sanctions.

If an agent has a set of candidate actions subject to a set of conflict-free norms, then predicate *inScope* can be used to select among the actions, namely those that are not within the scope of any prohibitions. Alternatively, agents can use the mechanism above to select those actions that are within the scope of obligations, and hence should be given priority. These strategies have been explored in [6].

3 Contracts as Rules for Managing Enactment States

In this section we introduce a rule-based language for the explicit management of events generated by agents and the effects they cause – we introduced alternative versions of this formalism in [7, 8]: rules depict how norms should be inserted and removed as a result of agents' actions. A contract is a set of such rules, specifying how agents' normative positions change as a result of their actions.

For our computational model we propose a global account of all actions performed, as well as all norms which currently hold. We make use of the set $\varDelta$ to store action tuples and norms – it represents a *trace* or a history of the enactment of a society of agents from a global point of view:

Definition 9. *A global enactment state $\varDelta$ is a finite, possibly empty, set of action tuples $\langle a\!:\!r, \bar{\varphi}\rangle$ and norms ω.*

A global enactment state $\varDelta$ can be "sliced" into many partial states $\varDelta_a = \{\langle a\!: r, \bar{\varphi}\rangle \in \varDelta \,|\, a \in \mathit{Agents}\}$ containing all actions of a specific agent a. Similarly, we could have partial states $\varDelta_r = \{\langle a : r, \bar{\varphi}\rangle \in \varDelta \,|\, r \in \mathit{Roles}\}$, representing the global state $\varDelta$ "sliced" across the various roles. We make use of a global enactment state to simplify our exposition; however, a fully distributed (and thus more scalable) account of enactment states can be achieved by slicing them as above and managing them in a distributed fashion[1].

Figure 2 depicts how our computational model works. An initial enactment state $\varDelta_0$ (possibly empty) is offered (represented by "$\Rrightarrow$") to a set of agents $(ag_1, \ldots, ag_n)$. These agents can add their events $(\varXi_1^0, \ldots, \varXi_n^0)$ to the state of affairs (via "$\updownarrow$"). $\varXi_i^j$ is the (possibly empty) set of events added by agent i at state of affairs $\varDelta_j$. After an established amount of time, we perform an exhaustive application of rules (denoted by "$\overset{*}{\rightsquigarrow}$") to the enactment state $\varDelta_0 \cup \varXi_1^0 \cup \cdots \cup \varXi_n^1$, yielding a new enactment state $\varDelta_1$. This new state will, on its turn, be offered to the agents for them to add their events, and the same process will go on.

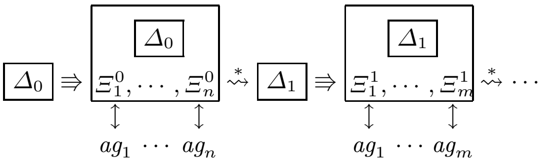

$\varDelta_0$
$\Rrightarrow$
$\varDelta_0$
$\varXi_1^0, \cdots, \varXi_n^0$
$\overset{*}{\rightsquigarrow}$
$\varDelta_1$
$\Rrightarrow$
$\varDelta_1$
$\varXi_1^1, \cdots, \varXi_m^1$
$\overset{*}{\rightsquigarrow}$ $\cdots$
ag_1 $\cdots$ ag_n
ag_1 $\cdots$ ag_m

Fig. 2. Semantics as a Sequence of $\varDelta$'s

3.1 A Rule Language for Managing Normative Positions

Our rules are constructs of the form $\mathit{LHS} \rightsquigarrow \mathit{RHS}$, where LHS contains a representation of parts of the current enactment state which, if they hold, will cause

[1] In [6] we present a distributed architecture for electronic institutions [4], in which global enactment states are broken down into *scenes*, that is, agent sub-activities with specific purposes, such as the registration process in a virtual auction room, the auction itself and the settlement of bills (and delivery of goods).

the rule to be triggered. *RHS* describes the updates to the current enactment state, yielding the next enactment state:

Definition 10. *A rule R is defined by the following grammar:*

$$\begin{aligned} R &::= \mathit{LHS} \rightsquigarrow \mathit{RHS} \\ \mathit{LHS} &::= \mathit{LHS} \wedge \mathit{LHS} \mid \neg \mathit{LHS} \mid \mathit{Action} \mid \omega \mid \gamma \\ \mathit{RHS} &::= \mathit{RHS} \wedge \mathit{RHS} \mid \oplus\, \omega \mid \ominus\, \omega \end{aligned}$$

Intuitively, the left-hand side *LHS* describes the conditions the current enactment state ought to have for the rule to apply. The right-hand side *RHS* describes the updates to be performed to the current enactment state, yielding the next enactment state.

3.2 Semantics of Rules

As suggested in Figure 2, we define the semantics of our rules as a relationship between the current enactment state and the next enactment state. In this section we define this relationship.

We first define the semantics of the *LHS* of a rule, that is, how a rule is triggered:

Definition 11. $\mathbf{s}_l(\Delta, \mathit{LHS}, \sigma)$ *holds between an enactment state* Δ*, the left-hand side of a rule LHS and a substitution* σ *depending on the format of LHS:*

1. $\mathbf{s}_l(\Delta, \mathit{LHS} \wedge \mathit{LHS}', \sigma)$ *holds iff* $\mathbf{s}_l(\Delta, \mathit{LHS}, \sigma')$ *and* $\mathbf{s}_l(\Delta, \mathrm{LHS}' \cdot \sigma', \sigma'')$ *hold,* $\sigma = \sigma' \cup \sigma''$.
2. $\mathbf{s}_l(\Delta, \neg\mathit{LHS}, \sigma)$ *holds iff* $\mathbf{s}_l(\Delta, \mathit{LHS}, \sigma)$ *does not hold.*
3. $\mathbf{s}_l(\Delta, \mathit{Action}, \sigma)$ *holds iff* $\mathit{Action} \cdot \sigma \in \Delta$.
4. $\mathbf{s}_l(\Delta, \omega, \sigma)$ *holds iff* $\omega \cdot \sigma \in \Delta$.
5. $\mathbf{s}_l(\Delta, \gamma, \sigma)$ *holds iff* $\mathit{satisfy}(\{\gamma \cdot \sigma\})$.

Case 1 depict the semantics of conjunctions and how their individual substitutions are combined. Case 2 introduces the negation by failure: a negated action is true if, and only if, it has not taken place, that is, it is not found in the enactment state. Case 3 holds when an action is found in the enactment state. Case 3 holds when a norm is found in the enactment state. Case 5 holds if a constraint is satisfiable, after applying a substitution σ to it.

We now define the semantics of the *RHS* of a rule:

Definition 12. *Relation* $\mathbf{s}_r(\Delta, \mathit{RHS}, \Delta')$ *mapping an enactment state* Δ*, the right-hand side of a rule RHS and a new enactment state* Δ' *is defined as:*

1. $\mathbf{s}_r(\Delta, \mathit{RHS} \wedge \mathit{RHS}', \Delta'')$ *holds iff* $\mathbf{s}_r(\Delta, \mathit{RHS}, \Delta')$ *and* $\mathbf{s}_r(\Delta', \mathit{RHS}', \Delta'')$ *hold.*
2. $\mathbf{s}_r(\Delta, \oplus\omega, \Delta \cup \{\omega\})$ *holds.*
3. $\mathbf{s}_r(\Delta, \ominus\omega, \Delta \setminus \{\omega\})$ *holds.*

Case 1 decomposes a conjunction and builds the new state by merging the partial states of each update. Case 2 caters for the insertion of norms and case 3 defines how a norm is deleted.

Our rules are *exhaustively* applied on the enactment states thus considering all matching atomic formulae. We thus need relationship $\mathbf{s}_l^*(\varDelta, \mathit{LHS}, \varSigma)$ which obtains in $\varSigma = \{\sigma_0, \ldots, \sigma_n\}$ *all* possible matches of the left-hand side of a rule:

Definition 13. $\mathbf{s}_l^*(\varDelta, \mathit{LHS}, \varSigma)$ *holds, iff* $\varSigma = \{\sigma_1, \ldots, \sigma_n\}$ *is the largest non-empty set such that* $\mathbf{s}_l(\varDelta, \mathit{LHS}, \sigma_i), 1 \leq i \leq n$, *holds.*

In the complete definition of the rule system, we define the semantics of our rules as relationships between enactment states: rules map an existing enactment state to a new enactment state. We adopt the usual semantics of production rules [14], that is, we exhaustively apply each rule by matching its *LHS* against the current state and use the values of variables obtained in this match to instantiate *RHS*.

3.3 An Interpreter for Contracts

The semantics above provides a basis for an interpreter for rules, shown in Fig. 3 as a logic program, interspersed with built-in Prolog predicates; for easy referencing, we show each clause with a number on its left. Clause 1 contains the top most definition: given a $\varDelta$ and a set of rules (a contract) *Rs*, it shows how we can obtain the next state $\varDelta'$ by finding (via the built-in `findall` predicate[2]) all those rules in *Rs* (picked by the `member` built-in) whose *LHS* holds in $\varDelta$ (checked via the auxiliary definition $\mathbf{s}_l^*$). This clause then uses the *RHS* of those rules with their respective sets of substitutions $\varSigma$ as the arguments of $\mathbf{s}'_r$ to finally obtain $\varDelta'$.

```
 1. $\mathbf{s}^*(\varDelta, Rs, \varDelta') \leftarrow$
      $\mathtt{findall}(\langle RHS, \varSigma\rangle, (\mathtt{member}((LHS \leadsto RHS), Rs), \mathbf{s}_l^*(\varDelta, LHS, \varSigma)), RHSs),$
      $\mathbf{s}'_r(\varDelta, RHSs, \varDelta')$

 2. $\mathbf{s}_l^*(\varDelta, LHS, \varSigma) \leftarrow \mathtt{findall}(\sigma, \mathbf{s}_l(\varDelta, LHS, \sigma), \varSigma)$
 3. $\mathbf{s}_l(\varDelta, (Action \wedge LHS), \sigma_1 \cup \sigma_2) \leftarrow \mathbf{s}_l(\varDelta, Action, \sigma_1), \mathbf{s}_l(\varDelta, LHS, \sigma_2)$
 4. $\mathbf{s}_l(\varDelta, \neg LHS, \sigma) \leftarrow \neg \mathbf{s}_l(\varDelta, LHS, \sigma)$
 5. $\mathbf{s}_l(\varDelta, Action, \sigma) \leftarrow \mathtt{member}(Action \cdot \sigma, \varDelta)$
 6. $\mathbf{s}_l(\varDelta, \omega, \sigma) \leftarrow \mathtt{member}(\omega \cdot \sigma, \varDelta)$
 7. $\mathbf{s}_l(\varDelta, \gamma, \sigma) \leftarrow \mathit{satisfy}(\{\gamma \cdot \sigma\})$

 8. $\mathbf{s}'_r(\varDelta, RHS, \varDelta') \leftarrow$
      $\mathtt{findall}(\varDelta'', (\mathtt{member}(\langle RHS, \varSigma\rangle, RHSs), \mathtt{member}(\sigma, \varSigma), \mathbf{s}_r(\varDelta, RHS \cdot \sigma, \varDelta'')), AllDeltas),$
      $\mathit{merge}(AllDeltas, \varDelta')$
 9. $\mathbf{s}_r(\varDelta, (U \wedge RHS), \varDelta_1 \cup \varDelta_2) \leftarrow \mathbf{s}_r(\varDelta, U, \varDelta_1), \mathbf{s}_r(\varDelta, RHS, \varDelta_2)$
10. $\mathbf{s}_r(\varDelta, \oplus\omega, \varDelta \cup \{\omega\})) \leftarrow$
11. $\mathbf{s}_r(\varDelta, \ominus\omega, \varDelta \setminus \{\omega\})) \leftarrow$
```

Fig. 3. An Interpreter for Contracts

Clause 2 implements $\mathbf{s}_l^*$: it finds all the different ways (represented as individual substitutions σ) that the left-hand side *LHS* of a rule can be matched in

[2] ISO Prolog built-in `findall/3` obtains all answers to a query (2nd argument), recording the values of the 1st argument as a list stored in the 3rd argument.

an enactment state Δ – the individual σ's are stored in sets Σ of substitutions, as a result of the `findall`/3 execution. Clauses 3-7 are adaptations of Def. 11.

Clause 8 shows how $\mathbf{s}'_r$ computes the new enactment state using the current enactment state and a list *RHSs* of pairs $\langle RHS, \Sigma \rangle$ (obtained in the second body goal of clause 1): it picks out (via predicate `member`/2) each individual substitution $\sigma \in \Sigma$ and uses it in *RHS* to compute via $\mathbf{s}_r$ a partial new state Δ'' which is stored in *AllDeltas*. *AllDeltas* contains a set of partial new states and these are combined together via the *merge*/2 predicate – it joins all the partial states, removing any replicated components. Clauses 9-11 are adaptations of Def. 12.

4 Norm Conflicts

This section provides definitions for norm conflicts, enabling their detection and resolution. Constraints confer more expressiveness and precision on norms, but mechanisms for detection and resolution must factor them in.

4.1 Conflict Detection

A conflict arises when an action is simultaneously prohibited and permitted/obliged, and its variables have overlapping values. The variables of a norm specify its scope of influence, that is, which agent/role the norm concerns, and which values of the action it addresses. In Fig. 4, we show two norms over action *deploy*(S, X, Y), establishing that sensor S is to be deployed on grid position (X, Y). The norms are

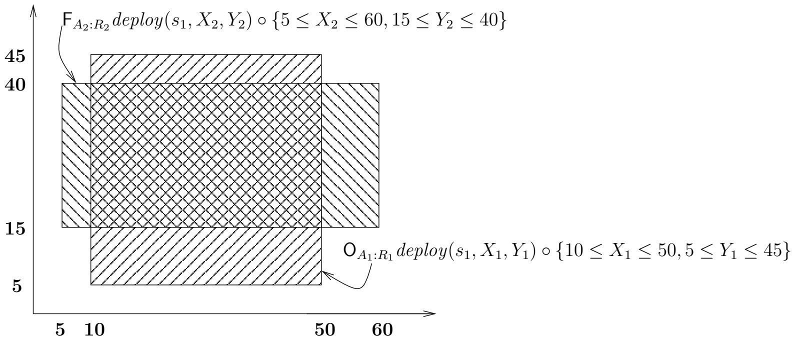

$\mathsf{F}_{A_2:R_2}\,deploy(s_1, X_2, Y_2) \circ \{5 \leq X_2 \leq 60, 15 \leq Y_2 \leq 40\}$
$\mathsf{O}_{A_1:R_1}\,deploy(s_1, X_1, Y_1) \circ \{10 \leq X_1 \leq 50, 5 \leq Y_1 \leq 45\}$
45
40
15
5
5
10
50
60

Fig. 4. Conflict Detection: Overlap in Scopes of Influence

$$\mathsf{O}_{A_1:R_1}\,deploy(s_1, X_1, Y_1) \circ \{10 \leq X_1 \leq 50, 5 \leq Y_1 \leq 45\}$$

$$\mathsf{F}_{A_2:R_2}\,deploy(s_1, X_2, Y_2) \circ \{5 \leq X_2 \leq 60, 15 \leq Y_2 \leq 40\}$$

Their scopes are shown as rectangles filled with different patterns. The overlap of their scopes is the rectangle in which both patterns are superimposed. Norm conflict is formally defined as follows:

Definition 14. *Norms $\omega, \omega' \in \Delta$, are in conflict under substitution σ, denoted as conflict$(\omega, \omega', \sigma)$, X being O or P, iff:*

- $\omega = \mathsf{F}_{\alpha:\rho}\varphi \circ \Gamma$, $\omega' = \mathsf{X}_{\alpha':\rho'}\varphi' \circ \Gamma'$ *or*
- $\omega = \mathsf{X}_{\alpha:\rho}\varphi \circ \Gamma$, $\omega' = \mathsf{F}_{\alpha':\rho'}\varphi' \circ \Gamma'$

and the following conditions hold:

1. *unify*$(\langle \alpha, \rho, \varphi \rangle, \langle \alpha', \rho', \varphi' \rangle, \sigma)$ *and*
2. *satisfy*$((\Gamma \cup \Gamma') \cdot \sigma)$

That is, a conflict occurs between a prohibition and either an obligation or a permission if *1)* a substitution σ can be found that unifies the variables of the two norms, and *2)* the constraints from both norms can be satisfied (taking σ under consideration).

The norm conflict of Fig. 4 is indeed captured by Definition 14. We can obtain a substitution $\sigma = \{X_1/X_2, Y_1/Y_2\}$ and this is a first indication that there may be a conflict or *overlap* of influence between both norms regarding the defined action. The constraints on the norms may restrict the overlap and, therefore, leave actions under certain variable bindings free of conflict. We, therefore, have to investigate the constraints of both norms in order to see if an overlap of the values indeed occurs. In our example, the obligation has constraints $\{10 \leq X_1 \leq 50, 5 \leq Y_1 \leq 45\}$ and the prohibition has constraints $\{5 \leq X_2 \leq 60, 15 \leq Y_2 \leq 40\}$. By using the substitutions we can "merge" the constraints as $\{10 \leq X_2 \leq 50, 5 \leq X_2 \leq 60, 5 \leq Y_2 \leq 45, 15 \leq Y_2 \leq 40\}$; the overlap of the merged constraints is $10 \leq X_2 \leq 60$ and $15 \leq Y_2 \leq 40$ and they represent ranges of values for variables X_1, X_2 and Y_1, Y_2 where a conflict will occur.

For convenience (and without any loss of generality), we assume that our norms are in a special format: all terms τ occurring in ω are replaced by a fresh variable x (not occurring anywhere in ω) and a constraint $x = \tau$ is added to Γ. This is an extended form of *explicit unification* [20] and the transformation of formulae from their usual format to this extended explicit unification format can be easily automated by scanning ω from left to right, collecting all terms $\{\tau_1, \ldots, \tau_n\}$; then we add $\{x_1 = \tau_1, \ldots, x_n = \tau_n\}$ to Γ. For example, norm $\mathsf{P}_{A:R}deploy(s_1, X, Y) \circ \{X > 50\}$ becomes $\mathsf{P}_{A':R'}deploy(S, X', Y') \circ \{A' = A, R' = R, S = s_1, X' = X, Y' = Y, X > 50\}$. Although some of the added constraints $x = y$ may seem superfluous, they are required to ensure that unconstrained variables are properly dealt by our conflict resolution mechanism presented below.

4.2 Conflict Resolution

We resolve conflicts by manipulating the constraints associated to the norms' variables, removing any overlap in their values. In Fig. 5 we show the norms of Fig. 4 without the intersection between their scopes of influence[3] – the prohibition has been *curtailed*, its scope being reduced to avoid the values that

[3] For clarity, in this example we show the norms in their usual format without explicit unifications.

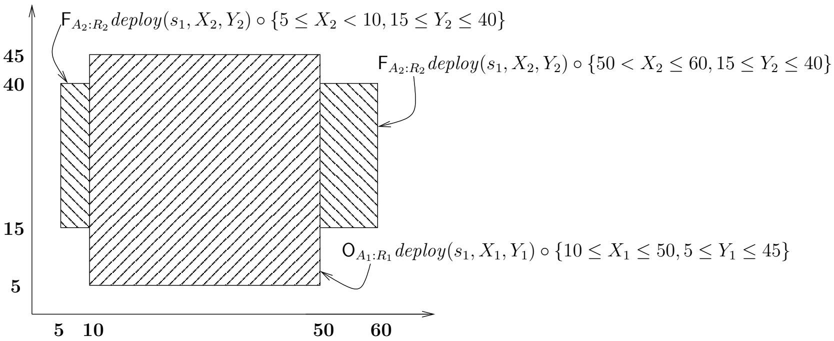


Fig. 5. Conflict Resolution: Curtailment of Scopes of Influence

the obligation addresses. Specific constraints are added to the prohibition in order to perform this curtailment; these additional constraints are derived from the obligation, as we explain below. In our example, we obtain two prohibitions, *viz.*, $\mathsf{F}_{A_2:R_2}deploy(s_1, X_2, Y_2) \circ \{5 \leq X_2 < 10, 15 \leq Y_2 \leq 40\}$ and $\mathsf{F}_{A_2:R_2}deploy(s_1, X_2, Y_2) \circ \{50 < X_2 \leq 60, 15 \leq Y_2 \leq 40\}$.

We formally define below how the curtailment of norms takes place. It is important to notice that the curtailment of a norm creates a new set Ω of curtailed norms:

Definition 15. *Relationship curtail*$(\omega, \omega', \Omega)$*, where*

- $\omega = \mathsf{X}_{\alpha:\rho}\varphi \circ \{\gamma_0, \ldots, \gamma_n\}$ *and*
- $\omega' = \mathsf{X}'_{\alpha':\rho'}\varphi' \circ \{\gamma'_0, \ldots, \gamma'_m\}$

X *and* X' *being either* O, F *or* P*, holds iff* Ω *is a possibly empty and finite set of norms obtained by curtailing* ω *with respect to* ω'*. The following cases arise:*

1. *If conflict*$(\omega, \omega', \sigma)$ *does not hold then* $\Omega = \{\omega\}$*; that is, the curtailment of a non-conflicting norm* ω *is* ω *itself.*
2. *If conflict*$(\omega, \omega', \sigma)$ *holds, then* $\Omega = \{\omega^c_0, \ldots, \omega^c_m\}$*, where* $\omega^c_j = \mathsf{X}_{\alpha:\rho}\varphi \circ (\{\gamma_0, \ldots, \gamma_n\} \cup \{\neg(\gamma'_j \cdot \sigma)\})$*,* $0 \leq j \leq m$*.*

In order to curtail ω, thus avoiding any overlapping of the values its variables may have with those variables of ω', we must "merge" the negated constraints of ω' with those of ω. Additionally, in order to ensure the appropriate correspondence of variables between ω and ω' is captured, we must apply the substitution σ obtained via *conflict*$(\omega, \omega', \sigma)$ on the merged negated constraints.

We combine the constraints of $\omega = \mathsf{X}_{\alpha:\rho}\varphi \circ \{\gamma_0, \ldots, \gamma_n\}$ with the negated constraints of $\omega' = \mathsf{X}'_{\alpha':\rho'}\varphi' \circ \{\gamma'_0, \ldots, \gamma'_m\}$. If we regard the set of constraints as a conjunction of constraints, that is, $\{\gamma_0, \ldots, \gamma_i\}$ is seen as $\bigwedge_{i=0}^{n} \gamma_i$, and if we regard "$\circ$" as the conjunction operator $\wedge$, then the following equivalences hold

$$\mathsf{X}_{\alpha:\rho}\varphi \wedge (\bigwedge_{i=0}^{n} \gamma_i \wedge \neg(\bigwedge_{j=0}^{m} \gamma'_j \cdot \sigma)) \equiv \mathsf{X}_{\alpha:\rho}\varphi \wedge (\bigwedge_{i=0}^{n} \gamma_i \wedge (\bigvee_{j=0}^{m} (\neg\gamma'_j \cdot \sigma)))$$

We can rewrite the last formula as

$$\bigvee_{j=0}^{m}(\mathsf{X}_{\alpha:\rho}\varphi \wedge (\bigwedge_{i=0}^{n}\gamma_i \wedge \neg(\gamma'_j \cdot \sigma)))$$

That is, each constraint on ω' leads to a possible solution for the resolution of a conflict and a possible curtailment of ω, as it prevents the overlap among variables. The curtailment thus produces a set of curtailed norms

$$\bigcup_{j=0}^{m}\omega_j^c = \bigcup_{j=0}^{m}\{\mathsf{X}_{\alpha:\rho}\varphi \circ (\{\gamma_0,\ldots,\gamma_n\} \cup \{\neg(\gamma'_j \cdot \sigma)\})\}$$

Although each of the $\omega_j^c, 0 \leq j \leq m$, represents a solution to the norm conflict, *all* of them are added to Ω in order to replace the curtailed norm. This allows the preservation of as much of the original scope of the curtailed norm as possible. Fig. 5 illustrates this: the result of the curtailment are two new prohibitions applicable to all those coordinates of the original prohibition which are not covered by the obligation, rather than just one of them. However, replacing the original prohibition with one of its curtailed versions would resolve the conflict.

5 Preemptive Normative Conflict Resolution

The rules of a contract create and remove norms. When new norms are introduced, they may conflict with each other. A post-conflict approach would invoke the norm curtailment mechanism above whenever a conflict arises during the enactment of a multi-agent system which is subject to a contract. We have pursued in [23] this approach: when a new norm is added to a set of norm, it is checked for conflicts and, depending on explicit policies, either the new norm is curtailed or existing norms are curtailed. When a norm is removed, any previous curtailments it caused on other norms are *undone*, this being achieved via a “roll back”/“roll forward” mechanism: the sequence (*i.e.*, the history) of all enactment states is maintained and we roll back to the state before the norm to be removed was introduced, then skip the following state and roll forward, introducing all the norms from that point onwards.

However, this approach is computationally very expensive, as we reported in [23]. We thus suggest a *preemptive approach* whereby rules are analysed beforehand for their potential conflicts, and then the norms appearing on their right-hand sides are curtailed, thus preventing any normative conflicts in the future.

Two rules R, R' have the potential for raising a normative conflict if *i)* their *RHS*s add conflicting norms to the enactment state and *i)* their *LHS*s can be simultaneously triggered. The first check is straightforward: we can scan the *RHS*s of the rules, collect their norms, and compare them two by two. The second check, however, is much trickier as we cannot in general decide if two *LHS*s *will be* simultaneously triggered: this check would require the exhaustive generation of all histories (*i.e.*, sequences of enactment states) and this could be prohibitively costly or, in the case of MASs which should run forever, impossible.

We address the second check in a conservative fashion: instead of checking whether the two *LHS*'s simultaneously trigger, we check if the situations they describe can possibly appear together. For instance, a if rule R has $\mathit{send}(Ag_1 : R_1, Ag_2 : R_2, \mathit{offer}(X))$ on its *LHS* and rule R' has $\neg\mathit{send}(Ag_1 : R_1, Ag_2 : R_2, \mathit{offer}(X))$ on its *LHS*, then we know for sure that these rules will never trigger simultaneously.

Given two rules $R = \mathit{LHS} \rightsquigarrow \mathit{RHS}$ and $R' = \mathit{LHS}' \rightsquigarrow \mathit{RHS}'$, we propose the $\mathit{compatible}(\mathit{LHS}, \mathit{LHS}')$ predicate to check if there could be an enactment state in which both *LHS* and LHS' holds. This predicate works by incrementally building an enactment state, adding to it all the actions, norms and constraints from *LHS* that are checked for, then extending the enactment state with the actions, norms and constraints that are checked for in LHS'. In this approach, we also add negated actions and norms to the state being built, so as to check if the state has a pair $\langle \mathit{Action}, \neg\mathit{Action}\rangle$, $\langle\omega, \neg\omega\rangle$ or $\langle\gamma, \neg\gamma\rangle$; if any of these appear in the state, then the rules are not compatible.

We not formally define the potential conflicts between two rules of a contract:

Definition 16. *Two rules $R = \mathit{LHS} \rightsquigarrow \mathit{RHS}$ and $R' = \mathit{LHS}' \rightsquigarrow \mathit{RHS}'$ are potentially in conflict, denoted as $\mathit{conflict}^r_p(R, R', \sigma)$, iff $\mathit{compatible}(\mathit{LHS}, \mathit{LHS}')$ holds, $\oplus\omega$ occurs in RHS, $\oplus\omega'$ occurs in RHS', and $\mathit{conflict}(\omega, \omega', \sigma)$.*

For the sake of simplicity, we assume in this paper that individual rules do not add conflicting norms. However, the mechanism we describe below could also be used to help engineers design individual rules, automatically spotting conflicts and suggesting changes to the rules.

5.1 Contract Formation via Preemptive Conflict Resolution

If two rules are found to be potentially in conflict, then their norm conflict(s) can be preemptively resolved by having the norms being added on their right-hand side curtailed using the mechanism presented above.

In this paper, we address the scenario in which two or more agents attempt to form a contract free from potential normative conflicts. The agents may have their own private contract(s) which they will need to take into account when forging new contracts. We consider two possible scenarios, explained below.

In the first scenario, an initiator agent ag_1 sends a proposal to another agent ag_2, consisting of a single rule R to become part of a contract between ag_1 and ag_2. Agent ag_2, the contacted agent, receives the rule and checks it against any of its current norms as well as rules of other contracts it has forged previously. Agent ag_2 then sends back a set $\mathcal{R}^c = \{R^c_1, \ldots, R^c_n\}$ of alternative versions of rule R, in which some of its added norms have been curtailed. The proposing agent ag_1 then chooses one of the rules from $\mathcal{R}^c$ and sends a message to ag_2 to inform its choice. This protocol is repeated again until the agents have a complete contract.

The other scenario is similar to the previous one, however, in addition to the set $\mathcal{R}^c = \{R^c_1, \ldots, R^c_n\}$ of alternative versions of the proposed rule, agent ag_2 also provides a *rationale* or justification for the suggestions. This rationale is in the form of ag_2's rules which have potential conflicts with R. When ag_1 receives the set $\mathcal{R}^c$ with the justifications, then (as in the previous scenario) it can accept the suggestions or, more interestingly, ag_1 may, on its turn, propose changes in the rules ag_2 used as rationale. This scenario may lead to longer interactions through which a new contract is forged via the revision of existing contracts. The revision of existing contracts use the same mechanism described here.

We present in Fig. 6 an algorithm which allows agents to analyse a proposed set of rules $\mathcal{R}$ with respect to another (pre-existing) set of rules $\mathcal{R}'$, providing a set of recommended changes $\mathcal{R}^c$ to $\mathcal{R}$. These changes will guarantee that there will be no normative conflicts when the agent takes part in enactments of multi-agent systems regulated by the contracts $\mathcal{R}^c$ and $\mathcal{R}'$

```
algorithm $preempt(\mathcal{R}, \mathcal{R}', \mathcal{R}^c)$
input a proposed set of new rules $\mathcal{R}$, and a set of old rules $\mathcal{R}'$
output a revised set of proposed rules $\mathcal{R}^c$
begin
  $\mathcal{R}^c \leftarrow \emptyset$
  for each $R' \in \mathcal{R}', R' = LHS' \rightsquigarrow RHS'$, do
  begin
    $conflict\_flag \leftarrow$ false
    $\mathcal{R}^t \leftarrow \emptyset$
    for each $R \in \mathcal{R}, R = LHS \rightsquigarrow RHS$, do
      if $conflict^r_p(R, R', \sigma)$ then
      begin
        $conflict\_flag \leftarrow$ true
        for each $\oplus\omega' \in \{RHS'\}$ do
          for each $\oplus\omega \in \{RHS\}$ do
          begin
            $curtail(\omega, \omega', \Omega)$
            for each $\omega^c \in \Omega$ do
            begin
              $\{RHS^t\} \leftarrow \{RHS\} \setminus \{\oplus\omega\} \cup \{\oplus\omega^c\}$
              $\mathcal{R}^t \leftarrow \mathcal{R}^t \cup \{LHS \rightsquigarrow RHS^t\}$
            end
          end
      end
    if $\neg conflict\_flag$ then
    begin
      $\mathcal{R}^c \leftarrow \mathcal{R}^c \cup \{R\}$
      $\mathcal{R} \leftarrow \mathcal{R} \setminus \{R\}$
    end
    else
      $\mathcal{R} \leftarrow \mathcal{R} \setminus \{R\} \cup \mathcal{R}^t$
  end
end
```

Fig. 6. Algorithm for Preemptive Contract Formation

In the algorithm we make use of $\{RHS\}$ to refer to the set with all the components of the right-hand side of the rule. More formally, we have:

Definition 17. *If* $RHS = \otimes\omega_1 \wedge \cdots \wedge \otimes\omega_n$ *(where* $\otimes$ *is either* $\oplus$ *or* $\ominus$*) then* $\{RHS\} = \{\otimes\omega_1, \ldots, \otimes\omega_n\}$.

The algorithm of Fig. 6 works by comparing each rule from $\mathcal{R}'$ with the candidate rules of $\mathcal{R}$. The algorithm makes use of a temporary set of changed rules $\mathcal{R}^t$ which is initialised to $\emptyset$, the empty set, in line 6 at the start of each loop with $R \in \mathcal{R}'$. The algorithm also makes use of a flag *conflict_flag* which is set true if there is a potential conflict between any rule $R' \in \mathcal{R}'$, that is, a rule from the set of old rules, and $R \in \mathcal{R}$ a new candidate rule – the flag is initialised to false in line 5 (with each new rule $R' \in \mathcal{R}'$), and switched to **true** in line 10, when a potential conflict is detected.

Loop 3–29 goes through the old rules $R' \in \mathcal{R}'$ and compares each of them with the new rules $R \in \mathcal{R}$ (loop 7–21). For each pair R, R' (respectively members from sets $\mathcal{R}'$ and $\mathcal{R}$), the algorithm checks for potential conflicts (line 8). If there is a potential conflict (cf. Def 16), lines 11-20 scan the RHS of both rules, obtaining a set Ω (line 14; cf. with Def. 14) which is the result of the curtailment of norm ω (from a new rule $R \in \mathcal{R}$) with respect to ω' (from an old rule $R' \in \mathcal{R}'$).

For each curtailed norm $\omega^c \in \Omega$ (lines 15–19), the algorithm creates an alternative RHS, replacing the old $\oplus\omega$ with $\oplus\omega^c$ (line 17). Line 18 updates the temporary set $\mathcal{R}^t$ of curtailed rules, adding the new rules obtained by replacing $\oplus\omega$ with $\oplus\omega^c$. Lines 22–28, executed after the loop of lines 7–21, update the set $\mathcal{R}^c$ of rules free from potential conflicts. Line 24 adds R to $\mathcal{R}^c$, since it is free from potential conflicts, and line 25 removes R from the set $\mathcal{R}$; if however, there has been a conflict, the set $\mathcal{R}$ should be updated, removing the conflicting R from it, and adding all the alternative formulations to R assembled in $\mathcal{R}^t$.

For the algorithm to work we make two assumptions. Firstly, we assume that the rules in $\mathcal{R}$ do not add norms with conflicts in their RHSs. Secondly, we assume that the rules in $\mathcal{R}$ do not have potential conflicts among themselves. The second assumption can be accommodated in a realistic setting if rules (possibly conflicting) are submitted one at a time to the algorithm; the first assumption requires the adaptation of the algorithm to address the design of rules, interleaving design with verification, a topic we elaborate further when we discuss future work below.

6 Related Work

In this section we refer to related work addressing aspects of contracts and normative systems.

A closely related work is that of [16]. The framework proposed in that paper includes contract specification, negotiation and monitoring, as well as the appropriate agent architecture to handle these aspects. However, in that paper issues of contract formation are not explored.

The work in [13] presents legal issues and surveys efforts at standardising contracts for electronic commerce. However, that paper does not propose a formal representation for contracts, nor does it investigate how contracts can be forged in an interactive way.

There are various other formulations for norms in the literature, as well as different ways to represent clauses of contracts using norms. In [8] we compare a rule-based formalism (notably more sophisticated than the one presented in this paper) with a number of alternative approaches. We show that it is possible to capture various normative phenomena with a rule-based formalism. Moreover, the rule-based formulation proves to be more compact, elegant, and intuitive than other formulations.

Our work is not concerned with contract negotiation based in game theoretical aspects, as explored in [2]. However, it has not escaped our attention that [2] employs a rule-based formalism, although their notion of norms is quite different from the one presented here; their conflicts are solved by means of priorities, which are used to choose a course of action.

Our approach to norm conflict detection and resolution can be contrasted with the work described in [11, 12]: the norms in their policies, although in an alternative syntax, have the same components as the norms presented in this paper, and hence the same expressiveness. However, conflicts are resolved in a coarser fashion: one of the conflicting norms is "overridden", that is, it becomes void. It is not clear how constraints in the norms of [11, 12] affect conflict, nor how conflicts are detected – from the informal explanation given, however, only direct conflicts are addressed. Our conflict resolution is finer-grained: norms are overridden for specific values (and not completely).

The work described in [3] analyses different normative conflicts – in spite of its title, the analysis is an informal one. That work differentiates between actions that are simultaneously prohibited and permitted – these are called *deontic inconsistencies* – and actions that are simultaneously prohibited and obliged – these are called *deontic conflicts*. The former is merely an "inconsistency" because a permission may not be acted upon, so no real conflict actually occurs. On the other hand, those situations when an action is simultaneously obliged and prohibited represent conflicts, as both obligations and prohibitions influence behaviours in an incompatible fashion. Our approach to detecting conflicts can capture the three forms of conflict/inconsistency of [18], *viz.* total-total, total-partial and intersection, respectively, when the permission entails the prohibition, when the prohibition entails the permission and when they simply overlap.

7 Summary, Conclusions and Future Work

We presented formal means to represent norms, that is, prohibitions, permissions, and obligations, and how these can be combined with a rule-based formalism to specify contracts. The left-hand side of our rule describe the circumstances which ought to arise for a norm to be revoked (removed) or introduced; the right-hand side of our rules specify which norms are to be revoked or introduced; we provided a simple semantics for our norms and rules.

Our norm representation uses constraints: these allow for a fine-grained control of the scope of the norm, that is, the values of the variables the norm refers to. These constraints are also useful when conflicts arise: we propose the resolution of normative conflicts via the careful manipulation of constraints. We provide means to detect normative conflicts and how to resolve them, and use these to propose a preemptive approach to contract formation, whereby agents exchange rules of a contract, checking these against any existing norms or other previously forged contracts. Our approach can be said to be preemptive because normative conflicts are considered *before* the contract is enacted and hence before any actual normative conflict arises.

We are exploring the proposed preemptive approach within the context of the ITA research project[4]. More specifically, we want to support coalition of human and software agents from disparate organisations (hence with different degrees of loyalty and willingness to share information and assets) to agree on the terms of a mission.

We want to adapt and extend the rule-based approach presented in this paper, using instead a logical approach. We envisage the clauses of a contract represented as formulae of a decidable fragment of first-order logic; in this approach a contract would be interpreted as a logical theory. Normative conflict can be detected via the reasoning mechanism of the logic, and the manipulation of constraints could still be used to resolve the conflicts.

References

1. K. R. Apt. *From Logic Programming to Prolog.* Prentice-Hall, U.K., 1997.
2. G. Boella and L. van der Torre. A Game Theoretic Approach to Contracts in Multiagent Systems. *IEEE Transactions on Systems, Man, and Cybernetics*, 36(1):68–79, Jan. 2006.
3. A. A. O. Elhag, J. A. P. J. Breuker, and P. W. Brouwer. On the Formal Analysis of Normative Conflicts. *Information & Comms. Techn. Law*, 9(3):207–217, Oct. 2000.
4. M. Esteva. *Electronic Institutions: from Specification to Development.* PhD thesis, Universitat Politècnica de Catalunya (UPC), 2003. IIIA monography Vol. 19.
5. M. Fitting. *First-Order Logic and Automated Theorem Proving.* Springer-Verlag, New York, U.S.A., 1990.
6. A. García-Camino, J.-A. Rodríguez-Aguilar, C. Sierra, and W. W. Vasconcelos. A Distributed Architecture for Norm-Aware Agent Societies. In M. Baldoni, U. Endriss, A. Omicini, and P. Torroni, editors, *Procs. of the 3rd Int'l Worskhop on Declarative Agent Languages and Technologies (DALT 2005), Selected and Revised Papers*, volume 3904 of *Lecture Notes in Computer Science*, pages 89–105. Springer-Verlag, Utrecht, The Netherlands, July 25, 2005, 2006.
7. A. García-Camino, J.-A. Rodríguez-Aguilar, C. Sierra, and W. W. Vasconcelos. A Rule-based Approach to Norm-Oriented Programming of Electronic Institutions. *ACM SIGecom Exchanges*, 5(5):33–40, Jan. 2006.
8. A. García-Camino, J.-A. Rodríguez-Aguilar, C. Sierra, and W. W. Vasconcelos. Constraint Rule-Based Programming of Norms for Electronic Institutions. *Journal of Autonomous Agents & Multiagent Systems*, 18(1):186–217, Feb. 2009.
9. J. Jaffar and M. J. Maher. Constraint Logic Programming: A Survey. *Journal of Logic Progr.*, 19/20:503–581, 1994.
10. J. Jaffar, M. J. Maher, K. Marriott, and P. J. Stuckey. The Semantics of Constraint Logic Programs. *Journal of Logic Programming*, 37(1-3):1–46, 1998.

[4] The International Technology Alliance (ITA) is a US/UK consortium of academic, military and industrial partners, pursuing research in technologies for knowledge management, communication and coordination among members of coalitions engaged in joint operations. Details about the ITA consortium are available at `http://www.usukita.org/`

11. L. Kagal and T. Finin. Modeling Communicative Behavior Using Permissions and Obligations. In *Lecture Notes in Computer Science*, volume 3396, pages 120–133, 2005.
12. L. Kagal and T. Finin. Modeling Conversation Policies using Permissions and Obligations. *Journal of Autonomous Agents & Multiagent Systems*, 14(2):187–206, Apr. 2007.
13. I. R. Kerr. Ensuring the Success of Contract Formation in Agent-Mediated Electronic Commerce. *Electronic Commerce Research*, 1(1-2):183–202, 2001.
14. B. Kramer and J. Mylopoulos. Knowledge Representation. In S. C. Shapiro, editor, *Encyclopedia of Artificial Intelligence*, volume 1, pages 743–759. John Wiley & Sons, 1992.
15. Z. Manna. *Mathematical Theory of Computation.* McGraw-Hill Kogakusha, Ltd., Tokio, Japan, 1974.
16. F. R. Meneguzzi, S. Miles, M. Luck, C. Holt, M. Smith, N. Oren, N. Faci, M. Kollingbaum, and S. Modgil. Electronic Contracting in Aircraft Aftercare: A Case Study. In M. Berger, B. Burg, and S. Nishiyama, editors, *Procs. 7th Int'l Joint Conf. on Autonomous Agents & Multiagent Systems (AAMAS 2008), Industry and Applications Track*, pages 63–70, Estorial, Portugal, May 2008. IFAAMAS.
17. O. Pacheco and J. Carmo. A Role Based Model for the Normative Specification of Organized Collective Agency and Agents Interaction. *Autonomous Agents and Multi-Agent Systems*, 6(2):145–184, 2003.
18. A. Ross. *On Law and Justice.* Stevens & Sons, 1958.
19. M. Sergot. A Computational Theory of Normative Positions. *ACM Trans. Comput. Logic*, 2(4):581–622, 2001.
20. L. Shapiro and E. Y. Sterling. *The Art of Prolog: Advanced Programming Techniques.* The MIT Press, April 1994.
21. Swedish Institute of Computer Science. *SICStus Prolog*, 2005. `http://www.sics.se/isl/sicstuswww/site/index.html`, viewed on 10 Feb 2005 at 18.16 GMT.
22. W. W. Vasconcelos, M. J. Kollingbaum, and T. J. Norman. Resolving Conflict and Inconsistency in Norm-Regulated Virtual Organizations. In *Procs. 6th Int'l Joint Conf. on Autonomous Agents & Multiagent Systems (AAMAS 2007)*, pages 632–639, Hawai'i, U.S.A., May 2007. IFAAMAS.
23. W. W. Vasconcelos, M. J. Kollingbaum, and T. J. Norman. Normative Conflict Resolution in Multi-Agent Systems. *Journal of Autonomous Agents & Multiagent Systems*, 2009. Volume and number to be confirmed. Available on-line at `http://www.springerlink.com/content/024242p7775530k6/`.

Distrust is not Always the Complement of Trust (Position Paper)

Célia da Costa Pereira
Università degli Studi di Milano, DTI
{pereira}@dti.unimi.it

Abstract. We believe that *distrust* can be as important as *trust* when agents are making a decision. An agent may not trust a source because of lack of positive evidence, but this does not necessarily mean the agent distrusts the source. Trust and distrust have to be considered as two separate concepts which can coexist. We are aware that an adequate way to take this fact into account is by considering explicitly not only the agent's degree of trust in a source but also its *independent* degree of distrust. Explicitly taking distrust into account allows us to mark a clear difference between the distinct notions of *negative* trust and *insufficient* trust. More precisely, it is possible, unlike in approaches where only trust is explicitly accounted for, to "weigh" differently information from *helpful*, *malicious*, *unknown*, or *neutral* sources.

1 Introduction and Motivations

Interaction is fundamental in a multi-agent system, allowing agents to cooperate to achieve their goals. When interchanging information, the extent to which a rational agent changes its beliefs may depend on several factors, like, for example, the trustworthiness of the agent providing new information, the agent's attitude towards information coming from unknown agents, or agents the agent knows as being malicious, or agents the agent knows as providers of usually correct information, and so on.

The main lack in most existing works on trust or using trust is the way the concept of *distrust* is implicitly considered, that is, as the complement of *trust* ($\textit{trust} = 1 - \textit{distrust}$). However, things are not always so simple. Trust and distrust may derive from different kinds of information (or from different sides of the personality) and, therefore, can coexist without being complementary [4, 9, 15]. For instance, one may not trust a source because of lack of positive evidence, but this does not necessarily mean (s)he distrusts it. Taking distrust explicitly into account allows an agent, e.g., to avoid dropping a goal just because favorable information comes from an unknown agent (neither trusted nor distrusted) — the absence of trust does not always mean full distrust.

We believe that an adequate way to take these facts into account is by considering explicitly not only the agent's trust degree in other agents but also its *independent* degree of distrust. The trustworthiness of a source can be represented as a (trust, distrust) pair, and intuitionistic fuzzy logic [1] can be used to represent the uncertainty on the trust degree introduced by the explicit presence of distrust.

Dagstuhl Seminar Proceedings 09121
Normative Multi-Agent Systems
http://drops.dagstuhl.de/opus/volltexte/2009/1915

Example John thinks his house has become too small for his growing family and would like to buy a larger one. Of course, John wants to spend as little money as possible. A friend who works in the real estate industry tells John prices are poised to go down. Then John reads in a newspaper that the real estate market is weak and prices are expected to go down. Therefore, John's desire is to wait for prices to lower before buying. However, John later meets a real estate agent who has an interesting house on sale, and the agent tells him to hurry up, because prices are soaring. On the way home, John hears a guy on the bus saying his cousin told him prices of real estate are going up.

We will see below which degrees could be assign by John to these different sources of information. Note that here, although numerical, the source degrees have only an ordinal significance.

2 Related Work: From Trust to Distrust

2.1 From Trust . . .

The term *trust* has a variety of meanings in the literature [14]. Demolombe [5] defines trust as a mental attitude of an agent with respect to another agent. He considers the agent's attitudes as a sort of belief about some property about another agent. He started by proposing a definition of trust as a binary concept (an agent trusts or does not trust another agent); then he introduced the notion of *graded trust* which is more suited to representing real situations where trust is perceived as less rigid. He also proposed a formal defintion for both *trust with respect to topics* and *conditional trust*.

Definition 1 (trust with respect to topics)
The fact that an agent a trust the agent b with respect to $prop$ for the topic t, $Tprop_{a,b}(t)$, is defined as follow:

$$Tprop_{a,b}(t) \equiv \forall\text{``}p''(A(t, \text{``}p'') \rightarrow Tprop_{a,b}(p),$$

where $A(t, \text{``}p'')$ means that the sentence named by “p'' is about the topic t.

The justification for this first proposal is that in general an agent trusts another agent for all the propositions related to a given topic.

Definition 2 (Conditional Trust)
The fact that an agent a trust the agent b for p in the circumstances represented by q, $Tprop_{a,b}(p|q)$, is defined as follow:

$$Tprop_{a,b}(p|q) \equiv K_a(q \rightarrow prop(p)),$$

where $prop(p)$ is any property about p.

The justification for this second proposal is that, in real life, there are many situations where an agent trusts another only in some particular circumstances. This is called for example, *context-awareness trust* [17] or *decision trust* [11].

Castelfranchi and Falcone [2] claimed that trust is much more than a subjective probability and pointed out to the necessity for a cognitive view of trust as a complex structure of beliefs (and goals) determining both a "degree of trust" (instead of a simple probability factor) and an estimation of risk. However, they expressed their awareness that in some situations, it should be important to consider the absolute values of some parameters independently from the values of the others. They present the following justification to this claim [7]: *"For example it is possible that the value of the damage per se (in case of failure) is too high to choose a given decision branch, and this independently either from the probability of the failure (even if it is very low) or from the possible payoff (even if it is very high). In other words, that danger might seem to the agent an intolerable risk"* .

Inspired by Castelfranchi and Falcone, Jøsang and colleagues [11] argued that an explicit distintion between *context-independent trust*, (which they called *reliability trust*), and *context-dependent trust*, (which they called *decision trust*), should be done when using the term trust. They adopt the definition proposed by Gambetta [8] as the definition of *reliability trust*.

Definition 3 (Reliability Trust)
Reliability trust is the subjective probability by which an individual, A, expects that another individual, B, performs a given action on which its welfare depends;

and before introducing the concept of *decision trust*, they propose the following example which can help us to understand the difference between *reliability trust* and *decision trust*:

Consider a person who distrusts an old rope for climbing from the third floor of a house during a fire exercise. Imagine now that the same person is trapped in a real fire in the same house, and that the only escape is to climb from the third floor window with the same old rope. In a real fire, most people would trust the rope. Although the reliability trust in the rope is the same in both situations, the decision trust changes as a function of the utility values associated with the possible courses of action.

Decision trust is then defined as:

Definition 4 (Decision Trust)
Decision trust is the extent to which a given party is willing to depend on something or somebody in a given situation with a feeling of relative security, even though negative consequences are possible.

The relation between trust and risk in decision making is also considered by Jøsang and Lo Presti in [10]. They propose one of the first models for trust in which a relation between *trust* and *risk* is considered explicitely. Their paper analyses the relationship between the two concepts by first looking at how a decision is made to enter into a transaction based on the risk information.

2.2 . . . to Distrust

While some researchers believe distrust simply means a low level of trust, others believe distrust is a concept entirely separate from trust. Here, we propose a non-exhaustive description of some works which share to some extent our viewpoint on trust and distrust.

Lewicki and colleagues [12] proposed a theoretical framework for understanding simultaneous trust and distrust within relationships. They assert that both trust and distrust involve movements toward certainty: trust concerning expectations of things hoped for and distrust concerning expectations of things feared. Like us, they belong to the thinking school which considers that *trust and distrust are separate but linked dimensions*, but not necessary the opposite ends of a single continuum. Indeed, the elements which contribute to the growth and decline of trust can be different from those which contribute to the growth and decline of distrust.

Griffiths [9] shown how agents can use trust to manage risk when cooperating. He proposed an approach which (i) uses fuzzy logic to represent trust and distrust; and (ii) allows agents to reason with uncertain and imprecise information regarding other's trustworthiness. He claimed that distrust is not simply a negation of trust, but rather, an explicit belief that an agent will act against the best interest of another. This is in line with Lewicki and colleagues' opinion and also with our opinion. The difference from our opinion is that we consider that distrust should not be always perceived as a reason to necessarily associate malicious intentions to the trustee. In case of interaction with a neutral trustee, that is, when the weight of the trustor's reasons to trust is the same as the weight of the trustor's reasons to distrust, automatically associating malicious (or helpful) purposes to the trustee would not be fair. In that case, decisions should be taken based upon other parameters. For example, an optimistic trustor would underestimate the reasons to distrust: *"an optimist is one who will look for the best in those with whom s(he) interacts"*[13]. A pessimistic trustor instead, would underestimate the reasons to trust.

McKnight and Cheverny [15] argued that trust and distrust are separate constructs that may exist simultaneously. They claimed that *"distrust is not only important because it allows one to avoid negative consequences, but because general distrust of other people and institutions is becoming more prevalent which means that it may, to an extent, be displacing trust as a social mechanism for dealing with risk. Indeed, under certain conditions, distrust may already be more useful or beneficial than trust."* They underline that *"without properly defining trust and distrust, it would be hard to tell which is more important and when."*

3 Towards an Explicit Representation of Distrust

Existing computational models usually deal with trust in a binary way: they assume that a source is either to be trusted or not, and they compute the probability that the source can be trusted. However, sources can not always be divided into trustworthy and untrustworthy in a clear-cut way. Some sources may be trusted to a certain extent. To take this fact into account, we think that trust and distrust should be represented as fuzzy degrees.

3.1 Basic Considerations

Fuzzy sets, introduced by Zadeh [19], are a generalization of classical sets obtained by replacing the characteristic function of a set A with a *membership function* μ_A, which can take up any value in $[0, 1]$. Let X be the universe of discourse and $x \in X$. The value $\mu_A(x)$ or, more simply, $A(x)$ is the membership degree of element x in A, i.e., the degree to which x belongs in A.

In [1], Atanassov extended the fuzzy set theory by introducing Intuitionistic Fuzzy Set (IFS for short) theory. In fuzzy set theory, it is implicitely assumed that the fact that an element x "belongs" with a degree $\mu_A(x)$ in a fuzzy set A, follows that x should "not belong" to A to the extent $1 - \mu_A(x)$. An intuitionistic fuzzy set F, instead, explicitely assigns to each element x of the considered universe both a degree of membership $\mu_F(x) \in [0, 1]$ and one of non-membership $\nu_F(x) \in [0, 1]$ which are such that:

$$\mu_F(x) + \nu_F(x) \leq 1.$$

Obviously, when $\mu_F(x) + \nu_F(x) = 1$ for all the elements of the universe, the traditional fuzzy set concept is recovered.

Deschrijver and Kerr showed in [6] that IFS theory is formally equivalent to Interval Valued Fuzzy Set (IVFS) theory which is another extention of fuzzy set theory in which the membership degrees are subintervals instead of numbers from $[0, 1]$ [18]. The IFS pair $(\mu_F(x), \nu_F(x))$ corresponds to the IVFS interval $[\mu_F(x), 1 - \nu_F(x)]$, indicating that the degree with which x "belongs in F is ranged from $\mu_F(x)$ to $1 - \nu_F(x)$. They defined the *hesitation degree*, $h \in [0, 1]$, as the length of such an interval. It is given by $h = 1 - \mu_F(x) - \nu_F(x)$. The longer the interval, the more doubt about the actual $\mu_F(x)$ value.

3.2 The Trustworthiness of a Source

The *trustworthiness* of a source (or of another agent) may be defined as, [4]:

Definition [Trustworthiness of a Source]
Let $t \in [0, 1]$ be the degree of trust the agent has in a source, and $d \in [0, 1]$ be its degree of distrust in the same source. The trustworthiness of that source for the agent is represented by pair (t, d), whith $t + d \leq 1$.

Following Deschrijver and Kerr's viewpoint, the trustworthiness (t, d) of a source corresponds to the interval $[t, 1 - d]$, indicating that the trust degree can range from t to $1 - d$. Therefore, the hesitation degree $h = 1 - t - d$ represents the uncertainty, or doubt, about the actual trust value. E.g., if a source has trustworthiness $(0.2, 0)$, this means that the agent trusts the source to degree 0.2, but possibly more, because there is much doubt ($h = 0.8$). More precisely, it means that the agent may trust the source to a degree varying from 0.2 to 1. Instead, if the trustworthiness is $(0.6, 0.4)$, the agent trusts the source to degree 0.6 but not more ($h = 0$).

Thanks to these considerations, we can represent the trustworthiness of a source more faithfully than as it is proposed in existing approaches. For example, we can explicitly represent the following cases of trustworthiness:

$(0, 1)$: the agent has reasons to fully distrust the source, hence it has no hesitation ($h = 0$),

$(0, 0)$: the agent has no information about the source and hence no reason to trust the source, but also no reason to distrust it; therefore, it fully hesitates in trusting it ($h = 1$),

$(1, 0)$: the agent has reasons to fully trust the source, hence it has no hesitation ($h = 0$).

As we can see, by considering both (and not neccessarily related concepts) trust and distrust, it is possible to differentiate between absence of trust caused by presence of distrust (e.g., information provided by a malicious source) versus by lack of knowledge (e.g., as towards an unknown source).

The sources can be classified in:

- *helpful source*: a source for which the reasons to believe in are stronger than the reasons to reject its information;
- *malicious source*: a source for which the reasons to reject its information are stronger than the reasons to believe it;
- *unknown source*: a source which never provided information to the agent before;
- *neutral source*: a source which provided to the agent as much true information as false.

Example Contiunued To sum up, John got information from four sources with different scores. The first source is friendly and competent; therefore, its score is $(1, 0)$. The second is supposedly competent and hopefully independent: therefore, its score might be something like $(\frac{1}{2}, \frac{1}{4})$. The third source is unknown, but has an obvious conflict of interest; therefore John assigns it a score of $(0, 1)$. Finally, the guy on the bus is a complete stranger reporting the opinion of another complete stranger. Therefore, its score cannot be other than $(0, 0)$.

4 Summary and Perspectives for a Normative Multiagent System

Taking distrust explicitly into account can help when making decisions in a situation where the agents are *collaborative*, that is, those which are considered as helpful sources; *wary*, which are suited to contexts where competition is the main theme; and *utility-driven*, for which a gain corresponds to a loss for its counterparts.

It would be interesting to take these considerations into account in the case of a Normative Multiagent System, where the behaviour of an agent depends on its internal components but also on the society it is part of.

Luck and colleagues [16], for example, proposed to analyse the agent's behaviour (reasoning) thanks to a three-dimensional space model, Figure 1, with motivations (axis x), norms (axis y), and trust (axis z). Each vertex in the space represents a kind of society. In particular, an increase in the value of x represents a prevalence of malicious motivations, indicating that agents are more likely to defect if they see more utility in alternative interactions; an increase in the value of y indicates the prevalence of stricter norms and enforcement which can constraint the motivations of agents and prevent them from acting maliciously if they intend to do so; finally, an increase in the value of z indicates an increase in the trust that agents place in other agents and, therefore, an increase in willingness to cooperate with others.

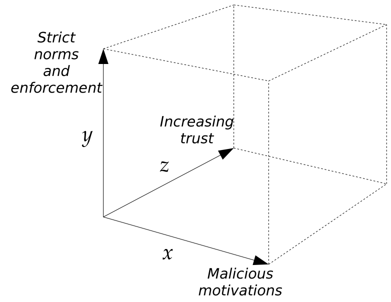


Fig. 1. Three-dimensional space model with norms, motivations and trust.

References

1. K T Atanassov. Intuitionistic fuzzy sets. *Fuzzy Sets Syst.*, 20(1):87–96, 1986.
2. C. Castelfranchi and R. Falcone. Trust is Much More Than Subjective Probability: Mental Components and Sources of Trust. In *HICSS'00*. 2000.
3. P. Cofta. Distrust. In *ICEC'06*, pages 250–258. 2006.
4. M. De Cock and P. Pinheiro da Silva. A Many Valued Representation and Propagation of Trust and Distrust. In *WILF'05*, pages 114-120. 2006.
5. R. Demolombe. Reasoning About Trust: A Formal Logical Framework. In *iTrust*, pages 291–303. 2004.
6. G. Deschrijver and E. E. Kerre. On the relationship between some extensions of fuzzy set theory. *Fuzzy Sets Syst.*, 133(2):227–235, 2003.
7. R. Falcone and C. Castelfranchi. Social Trust: A Cognitive Approach. In *Trust and Deception in Virtual Societibes*, pages 55–90. 2001.
8. D. Gambetta. Can we trust trust. In *Trust: Making and Breaking Cooperative Relations*, pages 213–237. 1988.
9. N. Griffiths. A fuzzy approach to reasoning with trust, distrust and insufficient trust. In *CIA*, pages 360–374. 2006.
10. A. Jøsang and S. Lo Presti. Analysing the Relationship Between Risk and Trust. In *Proc. of iTrust'04*, pages 135–145. 2004.
11. A. Jøsang and C. Keser and T. Dimitrakos. Can We Manage Trust. In *Proc. of iTrust'05*, pages 93–107. 2005.
12. R.J. Lewicki and D.J. McAllister, and R.J. Bies. Trust and Distrust: New Relationships and Realities. pages 23(3):438–458. 1998.
13. S. Marsh. Optimism and pessimism in trust. In *Proc. of IBERAMIA'94*, 1994.
14. D. H. McKnight and N. L. Chervany. The Meanings of Trust. University of Minnesota, Management Information Sytems Recherch Center. 1996.
15. D. H. McKnight and N. L. Chervany. Trust and Distrust Definitions: One Bite at a Time. In *Proceedings of the workshop on Deception, Fraud, and Trust in Agent Societies*, pages 27–54. 2001.
16. M. Luck and S. Munroe and F. Lopez y Lopez and R. Ashri. Trust and Norms for Interaction. In *Proc. of the IEEE International Conference on Systems, Man & Cybernetics*, pages 1944–1949. 2004.
17. S. Toivonen and G. Lenzini and I. Uusitalo Context-aware trustworthiness evaluation with indirect knowledge. In *In Proc. of 2nd International Semantic Web Policy Workshop (SWPW'06)*, 2006.

18. I. B. Türksen. Interval valued fuzzy sets based on normal forms. *Fuzzy Sets Syst.*, 20(2):191–210, 1986.
19. L. A. Zadeh. Fuzzy sets. *Information and Control*, 8:338–353, 1965.

Dynamic Context Logic and its Application to Norm Change

Guillaume Aucher[1], Davide Grossi[2], Andreas Herzig[3], Emiliano Lorini[3]
[1]University of Luxembourg
[2]University of Amsterdam
[3]Institut de Recherche en Informatique de Toulouse, France

Abstract

Building on a simple modal logic of context, the paper presents a dynamic logic characterizing operations of contraction and expansion on theories. We investigate the mathematical properties of the logic, and use it to develop an axiomatic and semantic analysis of norm change in normative systems. The proposed analysis advances the state of the art by providing a formal semantics of norm-change which, at the same time, takes into account several different aspects of the phenomenon, such as permission and obligation dynamics, as well as the dynamics of classificatory rules.

1 Introduction

Normative systems [4] have become a valuable abstraction for the design of multi-agent systems, and logic-based studies of norms have obtained increasing attention, in particular for their usefulness in providing computational models of norm-based interaction grounded on logical semantics (e.g. [1]). Taking up on pioneering work such as [3], the topic of how norms change over time has also become a topic of interest (e.g. [8]) given its relevance for understanding the ways social interaction evolves within multi-agent systems.

The aim of this work is to study norm change as a special instance of context change. Following [9] normative systems are, in a nutshell, logical theories concerning complex ways of classifying states of affairs as legal or illegal. As a consequence, each normative system specifies a context with respect to which rules of classification hold. Once such perspective is assumed, existing formal accounts of belief and knowledge dynamics can be transferred to study context change and, thus, norm change. We focus on dynamic epistemic logic (DEL) [17], and study two specific context change operations which can successfully account for norm change:

- **Context expansion** acounting for **norm promulgation**,
- **Context contraction** accounting for **norm derogation**.

Dagstuhl Seminar Proceedings 09121
Normative Multi-Agent Systems
http://drops.dagstuhl.de/opus/volltexte/2009/1900

In both cases, it is assumed that the authority of a normative system makes a proclamation in such a way that the norms of the normative system are modified. In the former case, the authority proclaims that from now on "a certain fact φ implies a violation", expanding the current set of obligations of the normative system. For example, the authority of a normative system might proclaim that from now on "driving faster than 110 km/h on a highway implies a violation". After this norm promulgation, it is obligatory to drive at most 110 km/h. In the latter case, the authority proclaims that "a certain fact φ does not imply a violation", contracting the current set of obligations of the normative system (and consequently making the normative system more 'permissive'). For example, the authority of a country might proclaim that from now on 'encrypting email does not imply a violation' by derogating the previous norm which forbade encryption in written communication. After this proclamation, it is permitted to encrypt email.

We start from the modal logic presented in [9]. This logic is based on a set of modal operators $[X]$ where X is a label denoting the context of a theory, i.e., in our case, the context of a normative system. A formula $[X]\varphi$ reads 'in the context of normative system X it is the case that φ'. Our aim in this paper is to extend this logic with two special kinds of events of the form $X{+}\psi$ and $X{-}\psi$, and corresponding modal operators $[X{+}\psi]$ and $[X{-}\psi]$. The former are similar to the operators for announcement studied in DEL [17]. Their function is to restrict the space of possible worlds accepted by the normative system X to the worlds where ψ is true. We use these operators to model norm promulgation. The function of modal operators of type $[X{-}\psi]$ is to add to the space of possible worlds accepted by the normative system X some worlds in which ψ is false. We use them to model norm derogation.

The paper is organized as follows. In Section 2 we will briefly present the modal logic of context of [9]. Section 3 is devoted to extend this logic with the two events $X{+}\psi$ and $X{-}\psi$ which allow to model context dynamics. Finally, in Section 4, we will apply our logical framework to norm change, *i.e.*norm promulgation and norm derogation.

2 A modal logic of context

The logic presented in this section is a simple modal logic designed to represent and reason about a localized notion of validity, that is, of validity with respect to all models in a given set. Such a given set is what is here called a *context*, in accord with much literature in artificial intelligence and linguistics on context theory (see, for instance, [14, 7]).

Let $\Phi = \{p, q, \ldots\}$ be a countable non-empty set of propositional letters, and let $\mathcal{C} = \{X, Y, \ldots\}$ be a countable set of contexts. $\mathcal{L}_{Prop}$ denotes the propositional language.

2.1 Models

Definition 1. *A context model (*$\mathbf{Cxt}$*-model) $\mathcal{M} = (W, R, \mathcal{I})$ is a tuple such that:*

- *W is a nonempty set of possible worlds;*

- *$R : \mathcal{C} \longrightarrow 2^W$ maps each context X to a subset of W;*
- *$\mathcal{I} : \Phi \longrightarrow 2^W$ is a valuation.*

We write R_X for $R(X)$ and $w \in \mathcal{M}$ for $w \in W$. For $w \in \mathcal{M}$, the couple $(\mathcal{M}, w)$ is a pointed context model.

A $\mathbf{Cxt}$-model represents a logical space together with some of its possible restrictions, i.e., the contexts. In our case, contexts are used to represent the restrictions to those sets of propositional models satisfying the rules stated by a given normative system [9]. Let us illustrate how they can be used to model normative systems.

Example 1. *Consider a normative system according to which: motorized vehicles must have a numberplate ; motorized vehicles must have an insurance; bikes should not have an insurance; bikes are classified as not being a motorized vehicle. Once a designated atom V is introduced in the language, which represents a notion of "violation" [5], the statements above obtain a simple representation:*

Rule 1: $(mt \wedge \neg pl) \rightarrow \mathsf{V}$

Rule 2: $(mt \wedge \neg in) \rightarrow \mathsf{V}$

Rule 3: $(bk \wedge in) \rightarrow \mathsf{V}$

Rule 4: $bk \rightarrow \neg mt$

A $\mathbf{Cxt}$-model $\mathcal{M} = (W, R, \mathcal{I})$ where $\mathcal{I}$ maps atoms mt, pl, in, bk and V to subsets of W models the normative system above as a context X if R_X coincides with the subset of W where Rules 1-4 are true according to propositional logic.

2.2 Logic

The logic $\mathbf{Cxt}$ is now presented which captures the notion of validity with respect to a context, thereby allowing to represent situations such as Example 1 in our language. To talk about $\mathbf{Cxt}$-models we use a modal language $\mathcal{L}_{\mathbf{Cxt}}$ containing modal operators $[X]$ for every $X \in \mathcal{C}$, plus the universal modal operator $[\mathsf{U}]$. The set of well-formed formulae of $\mathcal{L}_{\mathbf{Cxt}}$ is defined by the following BNF:

$$\mathcal{L}_{\mathbf{Cxt}} : \varphi ::= p \mid \neg\varphi \mid \varphi \wedge \varphi \mid [\mathsf{U}]\varphi \mid [X]\varphi$$

where p ranges over Φ and X over $\mathcal{C}$. The Boolean connectives $\top, \vee, \rightarrow, \leftrightarrow$ and the dual operators $\langle X \rangle$ are defined as usual within $\mathcal{L}_{\mathbf{Cxt}}$ as: $\langle X \rangle\varphi = \neg[X]\neg\varphi$, for $X \in \mathcal{C} \cup \{\mathsf{U}\}$.

We interpret formulas of $\mathcal{L}_{\mathbf{Cxt}}$ in a $\mathbf{Cxt}$-models as follows: the $[\mathsf{U}]$ operator is interpreted as the universal modality [6], and the $[X]$ operators model a restricted notion of validity.

Definition 2. *Let $\mathcal{M}$ be a $\mathbf{Cxt}$-model, and let $w \in \mathcal{M}$.*

$\mathcal{M}, w \models [X]\varphi$ *iff for all* $w' \in R_X$, $\mathcal{M}, w' \models \varphi$;
$\mathcal{M}, w \models [\mathsf{U}]\varphi$ *iff for all* $w' \in W$, $\mathcal{M}, w' \models \varphi$;
$\mathcal{M}, w \models p$ *iff* $w \in \mathcal{I}(p)$.

and as usual for the Boolean operators. Formula φ *is valid in* $\mathcal{M}$*, noted* $\mathcal{M} \models \varphi$*, iff* $\mathcal{M}, w \models \varphi$ *for all* $w \in \mathcal{M}$*.* φ *is* $\mathbf{Cxt}$*-valid, noted* $\models_{\mathbf{Cxt}} \varphi$*, iff* $\mathcal{M} \models \varphi$ *for all* $\mathbf{Cxt}$*-models* $\mathcal{M}$*.*

$\mathbf{Cxt}$-validity is axiomatized by the following schemas:

$(\mathtt{P})$	all propositional axiom schemas and rules
(4^{XY})	$[X]\varphi \to [Y][X]\varphi$
(5^{XY})	$\langle X\rangle\varphi \to [Y]\langle X\rangle\varphi$
$(\mathtt{T}^{\mathsf{U}})$	$[\mathsf{U}]\varphi \to \varphi$
$(\mathtt{K}^{X})$	$[X](\varphi \to \varphi') \to ([X]\varphi \to [X]\varphi')$
$(\mathtt{N}^{X})$	IF $\vdash \varphi$ THEN $\vdash [X]\varphi$

where $X, Y \in \mathcal{C} \cup \{\mathsf{U}\}$. The $[X]$ and $[Y]$ operators are $\mathbf{K45}$ modalities strengthened with the two inter-contextual interaction axioms 4^{XY} and 5^{XY}. $[\mathsf{U}]$ is an $\mathbf{S5}$ modality. Provability of a formula φ, noted $\vdash_{\mathbf{Cxt}} \varphi$, is defined as usual.

Logic $\mathbf{Cxt}$ is well-behaved from the point of view of both axiomatizability and complexity.

Theorem 1 ([9])**.** $\models_{\mathbf{Cxt}} \varphi$ *iff* $\vdash_{\mathbf{Cxt}} \varphi$.

Theorem 2. *Deciding* $\mathbf{Cxt}$*-validity is coNP-complete.*

Sketch of proof. Satisfiability of **S5** formulas is decidable in nondeterministic polynomial time [6]. Let $\mathcal{L}^{[\mathsf{U}]}$ be the language built from the set of atoms $\Phi \cup \mathcal{C}$ (supposing Φ and $\mathcal{C}$ are disjoint) and containing only one modal operator $[\mathsf{U}]$. That is:

$$\mathcal{L}^{[\mathsf{U}]} : \varphi ::= p \mid \neg\varphi \mid \varphi \wedge \varphi \mid [\mathsf{U}]\varphi$$

where p ranges over $\Phi\cup\mathcal{C}$. It gets a natural interpretation on context models where $[\mathsf{U}]$ is the global modality. Then one can show that the following is a satisfiability-preserving polytime reduction f of $\mathcal{L}_{\mathbf{Cxt}}$ to $\mathcal{L}^{[\mathsf{U}]}$: $f(p) = p$; $f(\neg\varphi) = \neg f(\varphi)$; $f(\varphi \wedge \varphi') = f(\varphi) \wedge f(\varphi')$; $f([\mathsf{U}]\varphi) = [\mathsf{U}]f(\varphi)$; $f([X]\varphi) = [\mathsf{U}](X \to f(\varphi))$. □

The same argument can be used to prove linear time complexity if the alphabet Φ is finite.

Another interesting property of $\mathbf{Cxt}$ is that every formula of $\mathcal{L}_{\mathbf{Cxt}}$ is provably equivalent to a formula without nested modalities, as the following proposition shows. We first formally define the language without nested modalities:

$$\mathcal{L}^1_{\mathbf{Cxt}} : \varphi ::= \alpha \mid [X]\alpha \mid [\mathsf{U}]\alpha \mid \neg\varphi \mid \varphi \wedge \varphi$$

where α ranges over $\mathcal{L}_{Prop}$ and X over $\mathcal{C}$.

Proposition 1. *For all* $\varphi \in \mathcal{L}_{\mathbf{Cxt}}$ *there is* $\varphi^1 \in \mathcal{L}^1_{\mathbf{Cxt}}$ *such that* $\vdash_{\mathbf{Cxt}} \varphi \leftrightarrow \varphi^1$.

Proof. By induction on φ. The Boolean cases clearly work. If φ is of the form $[X]\psi$ with $X \in \mathcal{C} \cup \{\mathsf{U}\}$ then by IH there are $\alpha_k, \alpha^i_j, \beta^i \in \mathcal{L}_{Prop}$ such that

$$\varphi \leftrightarrow [X] \bigwedge_{k\in\mathbb{N}_l} (\alpha_k \vee \bigvee_{i\in\mathbb{N}_{n_k}} ([X_i]\alpha^i_1 \vee \ldots \vee [X_i]\alpha^i_{n_i} \vee \langle X_i\rangle\beta^i))).$$

However, using (4^{XY}) and (5^{XY}), one can easily show that

$$\vdash_{\mathbf{Cxt}} [X](\alpha_k \vee \bigvee_{i\in\mathbb{N}_{n_k}} ([X_i]\alpha^i_1 \vee \ldots \vee [X_i]\alpha^i_{n_i} \vee \langle X_i\rangle\beta^i))) \leftrightarrow$$
$$([X]\alpha_k \vee \bigvee_{i\in\mathbb{N}_{n_k}} ([X_i]\alpha^i_1 \vee \ldots \vee [X_i]\alpha^i_{n_i} \vee \langle X_i\rangle\beta^i))).$$

□

We will use this result in the completeness proof of the dynamic extension of $\mathbf{Cxt}$ (Proposition 3).

2.3 Normative systems in $\mathbf{Cxt}$

We are ready to provide an object-level representation of Example 1. The contextual operators $[X]$ and the universal operator $[\mathsf{U}]$ can be used to define the concepts of *classificatory rule*, *obligation* and *permission* which are needed to model normative systems. Classificatory rules are of the form "φ counts as ψ in the normative system X" and their function in a normative systems is to specify classifications between different concepts [12]. For example, according to the classificatory rule "in the context of Europe, a piece of paper with a certain shape, color, *etc.*counts as a 5 Euro bill", in Europe a piece of paper with a certain shape, color, etc. should be classified as a 5 Euro bill. The concept of classificatory rule is expressed by the following abbreviation:

$$\varphi \Rightarrow_X \psi \overset{def}{=} [X](\varphi \rightarrow \psi)$$

where $\varphi \Rightarrow_X \psi$ reads 'φ counts as ψ in normative system X'. As done already in Example 1, by introducing the violation atom V we can obtain a reduction of deontic logic to logic $\mathbf{Cxt}$ along the lines first explored by Anderson [5]. As far as obligations are concerned, we introduce operators of the form $\mathbf{O}_X$ which are used to specify what is obligatory in the context of a certain normative system X:

$$\mathbf{O}_X\varphi \overset{def}{=} \neg\varphi \Rightarrow_X \mathsf{V}$$

According to this definition, 'φ is obligatory within context X' is identified with '$\neg\varphi$ counts as a violation in normative system X'. Note that we have the following $\mathbf{Cxt}$-theorem:

(1) $$\vdash_{\mathbf{Cxt}} ((\varphi \Rightarrow_X \psi) \wedge (\varphi \Rightarrow_X \neg\psi)) \rightarrow \mathbf{O}_X\neg\varphi$$

This will be of use in Section 4. Every $\mathbf{O}_X$ obeys axiom K and necessitation, and is therefore a normal modal operator.

(2) $$\vdash_{\mathbf{Cxt}} \quad \mathbf{O}_X(\varphi \rightarrow \psi) \rightarrow (\mathbf{O}_X\varphi \rightarrow \mathbf{O}_X\psi)$$

(3) $$\text{IF } \vdash_{\mathbf{Cxt}} \varphi \text{ THEN } \vdash_{\mathbf{Cxt}} \mathbf{O}_X\varphi$$

Note that the formula $\mathbf{O}_X\bot$ is consistent, hence our deontic operator does not satisfy the D axiom.

We define the permission operator in the standard way as the dual of the obligation operator: "φ is permitted within context X", noted $\mathbf{P}_X\varphi$. Formally:

$$\mathbf{P}_X\varphi \overset{def}{=} \neg\mathbf{O}_X\neg\varphi$$

$\mathbf{P}_{\mathsf{U}}\varphi$ should be read "φ is is deontically possible".

Example 2. *Consider again the normative system of Example 1. We can now express in* $\mathbf{Cxt}$ *that Rules 1-4 explicitly belong to context* X*:*

Rule 1′: $\mathbf{O}_X(mt \to pl)$

Rule 2′: $\mathbf{O}_X(mt \to in)$

Rule 3′: $\mathbf{O}_X(bk \to \neg in)$

Rule 4′: $bk \Rightarrow_X \neg mt$

Rules 1′-4′ explicitly localize the validity of Rules 1-4 of Example 1 to context X*. Logic* $\mathbf{Cxt}$ *is therefore enough expressive to represent several (possibly inconsistent) normative systems at the same time.*

The context representations enabled by $\mathbf{Cxt}$ are inherently static. The next section investigates context dynamics.

3 Dynamic context logic

3.1 Two relations on models

We first define the relations $\xrightarrow{X+\psi}$ and $\xrightarrow{X-\psi}$ on the set of pointed $\mathbf{Cxt}$-models.

Definition 3. *Let* $(\mathcal{M}, w) = (W, R, \mathcal{I}, w)$ *and* $(\mathcal{M}', w') = (W', R', \mathcal{I}', w')$ *be two pointed* $\mathbf{Cxt}$*-models, and let* $\varphi \in \mathcal{L}_{\mathbf{Cxt}}$ *and* $X \in \mathcal{C}$*.*
We set $(\mathcal{M}, w) \xrightarrow{X+\psi} (\mathcal{M}', w')$ *iff* $W = W', w = w', \mathcal{I} = \mathcal{I}'$*, and*

- $R'_Y = R_Y$ *if* $Y \neq X$*;*
- $R'_X = R_X \cap ||\psi||_{\mathcal{M}}$*.*

We set $(\mathcal{M}, w) \xrightarrow{X-\psi} (\mathcal{M}', w')$ *iff* $W = W', w = w', \mathcal{I} = \mathcal{I}'$*, and*

- $R'_Y = R_Y$ *if* $Y \neq X$*;*
- $R'_X = \begin{cases} R_X \text{ if } \mathcal{M}, w \models \neg[X]\psi \vee [\mathsf{U}]\psi \\ R_X \cup S \text{ otherwise, for some } \emptyset \neq S \subseteq ||\psi||_{\mathcal{M}} \end{cases}$

In case $(\mathcal{M}, w) \xrightarrow{X+\psi} (\mathcal{M}', w')$ *(resp.* $(\mathcal{M}, w) \xrightarrow{X-\psi} (\mathcal{M}', w')$*), we say that* $\mathcal{M}'$ *is a* (context) expansion *(resp.* contraction*) of* $\mathcal{M}$*.*

In the above definition, $||\psi||_{\mathcal{M}} = \{w \in \mathcal{M} : \mathcal{M}, w \models \psi\}$. So in both cases, it is only the context X which changes from $\mathcal{M}$ to $\mathcal{M}'$. In the first case, it is restricted to the worlds that satisfy ψ, and in the second case, it is enlarged with some worlds which satisfy $\neg\psi$, except if such worlds do not exist in the model ($[\mathsf{U}]\psi$) or if $\neg\varphi$ is already consistent with the context ($\neg[X]\psi$). Note that there might be several contractions of a given $\mathbf{Cxt}$-model but there is always a unique expansion. The relation $\xrightarrow{X-\psi}$ thus defines implicitly a *family* of contraction operations. The following proposition shows that $\xrightarrow{X-\psi}$ is essentially the converse relation of $\xrightarrow{X+\psi}$.

Proposition 2. *Let $(\mathcal{M}, w)$ and $(\mathcal{M}', w')$ be two pointed $\mathbf{Cxt}$-models and $\psi \in \mathcal{L}_{\mathbf{Cxt}}$. Then $(\mathcal{M}, w) \xrightarrow{X+\psi} (\mathcal{M}', w')$ iff*

$$(\mathcal{M}', w') \xrightarrow{X-\psi} (\mathcal{M}, w) \text{ and } \mathcal{M}', w' \models [X]\psi.$$

3.2 Logic

The language of the logic $\mathbf{DCxt}$ is obtained by adding the dynamic operators $[X{+}\psi]$ and $[X{-}\psi]$ to the language $\mathcal{L}_{\mathbf{Cxt}}$:

$$\mathcal{L}_{\mathbf{DCxt}} : \varphi ::= p \mid \neg\varphi \mid \varphi \wedge \varphi \mid [X]\varphi \mid [\mathsf{U}]\varphi \mid [X{+}\psi]\varphi \mid [X{-}\psi]\varphi$$

where p ranges over Φ, X over $\mathcal{C}$ and ψ over $\mathcal{L}_{\mathbf{Cxt}}$. $[X{+}\psi]\varphi$ reads 'after the expansion of the context X by ψ, φ is true', and $[X{-}\psi]\varphi$ reads 'after *any* contraction of the context X by ψ, φ is true'.

Definition 4. *Let $\mathcal{M}$ be a $\mathbf{Cxt}$-model. The truth conditions for $\mathcal{L}_{\mathbf{DCxt}}$ in $\mathcal{M}$ are those of Definition 2, plus:*

$\mathcal{M}, w \models [X{+}\psi]\varphi$ *iff* $\mathcal{M}', w' \models \varphi$ *for all* $\mathbf{Cxt}$-*models* $(\mathcal{M}', w')$
such that $(\mathcal{M}, w) \xrightarrow{X+\psi} (\mathcal{M}', w')$*;*
$\mathcal{M}, w \models [X{-}\psi]\varphi$ *iff* $\mathcal{M}', w' \models \varphi$ *for all* $\mathbf{Cxt}$-*models* $(\mathcal{M}', w')$
such that $(\mathcal{M}, w) \xrightarrow{X-\psi} (\mathcal{M}', w')$.

As before, $\mathcal{M} \models \varphi$ iff $\mathcal{M}, w \models \varphi$ for all $w \in \mathcal{M}$, and φ is $\mathbf{DCxt}$-valid ($\models_{\mathbf{DCxt}} \varphi$) iff $\mathcal{M} \models \varphi$ for all $\mathbf{Cxt}$-models $\mathcal{M}$.

The operator $[X{-}\psi]$ is thus useful if we want to have general properties about our family of contractions or about a situation; for example, given some formulas $\psi_1, \ldots, \psi_n$, what would be true after any sequence of contractions and expansions by these formulas? Can we get an inconsistency with a specific choice of contractions?

In order to axiomatize the $\mathbf{DCxt}$-validities we define for every $X \in \mathcal{C}$ two auxiliary languages $\mathcal{L}_{\neq X}$ and $\mathcal{L}_{=X}$:

$$\mathcal{L}_{=X} : \varphi ::= [X]\alpha \mid \neg\varphi \mid \varphi \wedge \varphi$$
$$\mathcal{L}_{\neq X} : \varphi ::= \alpha \mid [Y]\alpha \mid \neg\varphi \mid \varphi \wedge \varphi$$

where α ranges over $\mathcal{L}_{Prop}$ and Y over $(\mathcal{C} \cup \{\mathsf{U}\}) - \{X\}$.

Logic $\mathbf{DCxt}$ is axiomatized by the following schemata:

(Cxt) All axiom schemas and inference rules of $\mathbf{Cxt}$

(R+1) $[X{+}\psi]\varphi_{\neq X} \leftrightarrow \varphi_{\neq X}$

(R+2) $[X{+}\psi][X]\alpha \leftrightarrow [X](\psi \rightarrow \alpha)$

(R+3) $[X{+}\psi]\neg\varphi \leftrightarrow \neg[X{+}\psi]\varphi$

(R–1) $[X{-}\psi](\varphi_{\neq X} \vee \varphi_{=X}) \leftrightarrow (\varphi_{\neq X} \vee [X{-}\psi]\varphi_X)$

(R–2) $\neg[X{-}\psi]\bot$

(R–3) $$[X{-}\psi]([X]\alpha_1 \vee \ldots \vee [X]\alpha_n \vee \langle X\rangle\alpha) \leftrightarrow \\ ((\neg[X]\psi \vee [\mathsf{U}]\psi) \wedge ([X]\alpha_1 \vee \ldots \vee [X]\alpha_n \vee \langle X\rangle\alpha)) \\ \vee\, (([X]\psi \wedge \neg[\mathsf{U}]\psi) \wedge \\ ((\bigvee_i([X]\alpha_i \wedge [\mathsf{U}](\psi \vee \alpha_i))) \vee \langle X\rangle\alpha \vee [\mathsf{U}](\psi \vee \alpha)))$$

(K^{+}) $[X{+}\psi](\varphi \rightarrow \varphi') \rightarrow ([X{+}\psi]\varphi \rightarrow [X{+}\psi]\varphi')$

(K^{-}) $[X{-}\psi](\varphi \rightarrow \varphi') \rightarrow ([X{-}\psi]\varphi \rightarrow [X{-}\psi]\varphi')$

(RRE) Rule of replacement of proved equivalence

where $X \in \mathcal{C}$, $\varphi, \varphi' \in \mathcal{L}_{\mathbf{DCxt}}$, $\psi \in \mathcal{L}_{\mathbf{Cxt}}$, $\varphi_{=X} \in \mathcal{L}_{=X}$, $\varphi_{\neq X} \in \mathcal{L}_{\neq X}$, and $\alpha, \alpha_i \ldots \in \mathcal{L}_{Prop}$.

Note that from (R–1) and (R–2) one can deduce $[X{-}\psi]\varphi_{\neq X} \leftrightarrow \varphi_{\neq X}$. The above are reduction axioms:

Proposition 3. *For all $\varphi_{\mathbf{DCxt}} \in \mathcal{L}_{\mathbf{DCxt}}$ there is $\varphi_{\mathbf{Cxt}} \in \mathcal{L}_{\mathbf{Cxt}}$ such that $\vdash_{\mathbf{DCxt}} \varphi_{\mathbf{DCxt}} \leftrightarrow \varphi_{\mathbf{Cxt}}$.*

Sketch of proof. (By induction on the number of occurrences of dynamic operators.) Let $\varphi_{\mathbf{DCxt}} \in \mathcal{L}_{\mathbf{DCxt}}$ and $\varphi'_{\mathbf{DCxt}}$ be one of its sub-formulas of the form $[X{+}\psi]\varphi_{\mathbf{Cxt}}$ or $[X{-}\psi]\varphi_{\mathbf{Cxt}}$, with $\varphi_{\mathbf{Cxt}} \in \mathcal{L}_{\mathbf{Cxt}}$. By Proposition 1, there is $\varphi^1_{\mathbf{Cxt}} \in \mathcal{L}^1_{\mathbf{Cxt}}$ such that $\vdash_{\mathbf{Cxt}} \varphi_{\mathbf{Cxt}} \leftrightarrow \varphi^1_{\mathbf{Cxt}}$. So $\vdash_{\mathbf{DCxt}} [X{+}\psi]\varphi_{\mathbf{Cxt}} \leftrightarrow [X{+}\psi]\varphi^1_{\mathbf{Cxt}}$ by (REE) and (K^{+}). Now, thanks to axioms (R+1), (R+2) and (R+3) and because $\varphi^1_{\mathbf{Cxt}} \in \mathcal{L}^1_{\mathbf{Cxt}}$, one can easily show that there is $\psi_{\mathbf{Cxt}} \in \mathcal{L}_{\mathbf{Cxt}}$ such that $\vdash_{\mathbf{DCxt}} [X{+}\psi]\varphi^1_{\mathbf{Cxt}} \leftrightarrow \psi_{\mathbf{Cxt}}$. For the case $[X{-}\psi]\varphi_{\mathbf{Cxt}}$ we apply the same method using (R–1), (R–2) and (R–3). So $\vdash_{\mathbf{DCxt}} \varphi'_{\mathbf{DCxt}} \leftrightarrow \psi_{\mathbf{Cxt}}$. Now we replace $\varphi'_{\mathbf{DCxt}}$ by $\psi_{\mathbf{Cxt}}$ in $\varphi_{\mathbf{DCxt}}$. This yields an equivalent formula (thanks to (RRE)) with one dynamic operator less. We then apply to this formula the same process we applied to $\varphi_{\mathbf{Cxt}}$ until we get rid of all the dynamic operators. □

For example, $\vdash_{\mathbf{DCxt}} [X{-}\alpha]\neg[X]\alpha \leftrightarrow \langle\mathsf{U}\rangle\neg\alpha$. As in DEL, soundness and completeness follow from Proposition 3:

Theorem 3. $\models_{\mathbf{DCxt}} \varphi$ *iff* $\vdash_{\mathbf{DCxt}} \varphi$.

Theorem 4. *Deciding* $\mathbf{DCxt}$*-validity is decidable.*

Finally, we could perfectly enrich this formalism with specific contraction operators. For example we could add to $\mathcal{L}_{\mathbf{DCxt}}$ the contraction operator $[X \circeq \psi]\varphi$ whose semantics would be defined as follows: for $\mathcal{M} = (W, R, \mathcal{I})$, $\mathcal{M}, w \models [X \circeq \psi]\varphi$

iff $\mathcal{M}', w \models \varphi$, where $\mathcal{M}' = (W, R', \mathcal{I})$ with $R'_Y = R_Y$ for $Y \neq X$ and $R'_X = R_X \cup \{w \in W \mid \mathcal{M}, w \models \neg\psi\}$. To get a complete axiomatization, we just have to add to $\mathbf{DCxt}$ the following axiom schemas: (1) $[X \mathring{=} \psi]\varphi_{\neq X} \leftrightarrow \varphi_{\neq X}$; (2) $[X \mathring{=} \psi]\neg\varphi \leftrightarrow \neg[X \mathring{=} \psi]\varphi$; (3) $[X \mathring{=} \psi][X]\alpha \leftrightarrow [X]\alpha \wedge [\mathsf{U}](\neg\psi \rightarrow \alpha)$; and the distribution axiom ($\mathtt{K}^{\mathring{=}}$). In fact this contraction $\mathring{=}$ belongs to the family of contractions defined in Definition 3, and so we have $\vdash_{\mathbf{DCxt}} [X{-}\psi]\varphi \rightarrow [X \mathring{=} \psi]\varphi$.

4 A logical account of norm change

Just as we defined the static notions of obligation and classificatory rules on the basis of $\mathbf{Cxt}$, we can in the same spirit define the dynamic notions of promulgation and derogation of obligation and classificatory rules on the basis of $\mathbf{DCxt}$:

$$\begin{aligned} +(\varphi \Rightarrow_X \psi) &\overset{def}{=} X{+}(\varphi \rightarrow \psi) \\ +\mathbf{O}_X\psi &\overset{def}{=} X{+}(\neg\psi \rightarrow \mathsf{V}) \\ -(\varphi \Rightarrow_X \psi) &\overset{def}{=} X{-}(\varphi \rightarrow \psi) \\ -\mathbf{O}_X\psi &\overset{def}{=} X{-}(\neg\psi \rightarrow \mathsf{V}) \end{aligned}$$

$[+(\varphi \Rightarrow_X \psi)]\chi$ (resp. $[-(\varphi \Rightarrow_X \psi)]\chi$) should be read 'after the promulgation (resp. after *any* derogation) of the classificatory rule $\varphi \Rightarrow_X \psi$, χ is true'. Likewise, $[+\mathbf{O}_X\psi]\varphi$ (resp. $[-\mathbf{O}_X\psi]\varphi$) should be read 'after the promulgation (resp. after *any* derogation) within context X of the obligation ψ, χ is true'. Then we have the following intuitive $\mathbf{DCxt}$-theorems:

(4) $\vdash_{\mathbf{DCxt}} [+(\varphi \Rightarrow_X \psi)]\varphi \Rightarrow_X \psi$

(5) $\vdash_{\mathbf{DCxt}} [+\mathbf{O}_X\psi]\mathbf{O}_X\psi$

(6) $\vdash_{\mathbf{DCxt}} \mathbf{P}_{\mathsf{U}}\neg\psi \rightarrow [-\mathbf{O}_X\psi]\mathbf{P}_X\neg\psi$

In particular, $\mathbf{DCxt}$-theorem (6) says that "If $\neg\psi$ is deontically possible then after any derogation within context X of the obligation ψ, $\neg\psi$ is permitted".

Example 3. *In Example 2, after the legislator's proclamation that motorized vehicles having more than 50cc (mf) are obliged to have a numberplate (event $+\mathbf{O}_X((mt \wedge mf) \rightarrow pl)$ and that motorized vehicles having less than 50cc ($\neg mf$) are not obliged to have a numberplate (event $-\mathbf{O}_X((mt \wedge \neg mf) \rightarrow pl)$ we should expect that motorbikes having more than 50cc have the obligation to have a numberplate and motorbikes having less than 50cc have the permission not to have a numberplate. This is indeed the case:*

$$\vdash_{\mathbf{DCxt}} \mathbf{P}_{\mathsf{U}}(mt \wedge \neg mf \wedge \neg pl) \rightarrow ([+\mathbf{O}_X((mt \wedge mf) \rightarrow pl)]$$

$$[-\mathbf{O}_X((mt \wedge \neg mf) \rightarrow pl)]\mathbf{O}_X((mt \wedge mf) \rightarrow pl) \wedge$$

$$\mathbf{P}_X(mt \wedge \neg mf \wedge \neg pl)).$$

We now consider two types of normative inconsistency, classificatory dilemma and normative dilemma, and show how they might arise from promulgation and derogation.

Classificatory dilemna By classificatory dilemma we mean that a certain fact φ is classified by a normative system both under ψ and under $\neg\psi$, i.e. $(\varphi \Rightarrow_X \psi) \wedge (\varphi \Rightarrow_X \neg\psi)$. An example of classificatory dilemma is the case of someone who finds an object in the sea and is classified by the normative system as the owner of the object. At the same time, someone who claims having lost the object and can prove this, is also classified as the owner of the object. Finally, according to the normative system, there is no more than one owner of an object. If a person finds an object in the sea and another person claims that she has lost this object and can prove that, we incur a classificatory dilemma: the former person is classified as the owner of the object and, at the same time, she is classified as not being the owner of it.

Example 4. *In Example 2, after the legislator's proclamation that bikes with an engine must be classified as a motorized vehicles (event* $+\!\!|((bk \wedge en) \Rightarrow_X mt)$*), bikes with an engine are classified as motorized vehicles and, at the same time, they are classified as not being motorized vehicles. This is a classificatory dilemma:*

$$[+\!\!|((bk \wedge en) \Rightarrow_X mt)](((bk \wedge en) \Rightarrow_X mt) \wedge$$

$$((bk \wedge en) \Rightarrow_X \neg mt)).$$

Example 4 illustrates the following $\mathbf{DCxt}$-theorem:

$$\text{(7)} \qquad \begin{aligned} &\vdash_{\mathbf{DCxt}} (\varphi \Rightarrow_X \psi) \rightarrow [+\!\!|(\varphi \Rightarrow_X \neg\psi)] \\ &\qquad ((\varphi \Rightarrow_X \psi) \wedge (\varphi \Rightarrow_X \neg\psi)) \end{aligned}$$

The $\mathbf{Cxt}$-theorem (1) tells us that a classificatory dilemma implies $\mathbf{O}_X\neg\varphi$. It follows that if the normative system X is expanded with φ then $\bot$ becomes true in X, that is, the normative system becomes inconsistent:

$$\text{(8)} \qquad \vdash_{\mathbf{DCxt}} ((\varphi \Rightarrow_X \psi) \wedge (\varphi \Rightarrow_X \neg\psi)) \rightarrow [X{+}\varphi][X]\bot$$

Thus, changes generating classificatory dilemmas can be considered as badly designed normative modifications.

Normative dilemna By normative dilemma we mean a situation in which a normative system prescribes that a certain fact ψ must be true under a certain condition φ and at the same time $\neg\psi$ must be true under the same condition, i.e. $\mathbf{O}_X(\varphi \rightarrow \psi) \wedge \mathbf{O}_X(\varphi \rightarrow \neg\psi)$. An example of normative dilemma is the case of a soldier having at the same time the obligation to kill his enemies during a war and the obligation for every person not to shoot other people. If a soldier is classified as a person and enemies are classified as people, we incur a normative dilemma: a soldier has the obligation to shoot his enemies and the obligation not to shoot his enemies. Note that $\mathbf{O}_X\neg\varphi$ implies $\mathbf{O}_X(\varphi \rightarrow \psi) \wedge \mathbf{O}_X(\varphi \rightarrow \neg\psi)$ for every ψ. So, to be more precise, we should exclude from the previous definition of normative dilemna the situation in which $\mathbf{O}_X\neg\varphi$ holds.

Example 5. *In Example 2, after the legislator's proclamation that every bike must have an insurance (event* $+\!\!|\mathbf{O}_X(bk \rightarrow in)$ *), bikes have the obligation to have an insurance and the obligation not have it, which is a normative dilemma:*

$$[+\!\!|\mathbf{O}_X(bk \rightarrow in)](\mathbf{O}_X(bk \rightarrow in) \wedge \mathbf{O}_X(bk \rightarrow \neg in)).$$

Example 5 illustrates the following $\mathbf{DCxt}$-theorem:

$$\begin{aligned}\vdash_{\mathbf{DCxt}} \mathbf{O}_X(\varphi \to \psi) \to [{+}\mathbf{O}_X(\varphi \to \neg\psi)]\\ (\mathbf{O}_X(\varphi \to \psi) \land \mathbf{O}_X(\varphi \to \neg\psi))\end{aligned} \tag{9}$$

It is to be noted that, if a normative dilemma $\mathbf{O}_X(\varphi \to \psi) \land \mathbf{O}_X(\varphi \to \neg\psi)$ holds and the normative system is expanded with φ then every fact χ becomes obligatory in X:

$$\vdash_{\mathbf{DCxt}} (\mathbf{O}_X(\varphi \to \psi) \land \mathbf{O}_X(\varphi \to \neg\psi)) \to [X{+}\varphi]\mathbf{O}_X\bot \tag{10}$$

It is worth stressing the similarity between $\mathbf{DCxt}$-theorem (8) and $\mathbf{DCxt}$-theorem (10). While a classificatory dilemma results in an empty context ($\mathbf{DCxt}$-theorem (8)) under the assumption of the antecedent, a normative dilemma results in a context where legality is impossible ($\mathbf{DCxt}$-theorem (10)).

Finally, we have shown by $\mathbf{DCxt}$-theorems (7) and (9) that if we want to change a norm (a classificatory rule or an obligation) to a contrary norm by a sole act of norm promulgation we end up with a dilemma (either classificatory or normative). Thus, to avoid dilemmas, we must first derogate the old norm and then promulgate the contrary norm. This observation is formally expressed by the following $\mathbf{DCxt}$-theorems:

$$\begin{aligned}\vdash_{\mathbf{DCxt}} ((\varphi \Rightarrow_X \psi) \land \langle\mathsf{U}\rangle\neg(\varphi \to \psi)) \to\\ [{-}(\varphi \Rightarrow_X \psi)][{+}(\varphi \Rightarrow_X \neg\psi)]\\ \neg((\varphi \Rightarrow_X \neg\psi) \land (\varphi \Rightarrow_X \psi))\end{aligned} \tag{11}$$

$$\begin{aligned}\vdash_{\mathbf{DCxt}} (\mathbf{O}_X(\varphi \to \psi) \land \mathbf{P}_{\mathsf{U}}\neg(\varphi \to \psi)) \to\\ [{-}\mathbf{O}_X(\varphi \to \psi)][{+}\mathbf{O}_X(\varphi \to \neg\psi)]\\ \neg(\mathbf{O}_X(\varphi \to \neg\psi) \land \mathbf{O}_X(\varphi \to \psi))\end{aligned} \tag{12}$$

Note that by definition of $-$, these general results hold for *any* derogation (stemming from a contraction of Definition 3).

5 Related works

Formal models of norm change have been drawing attention since the seminal work of Alchourrón and Makinson on the logical structure of derogation in legal codes [3] which expanded into a more general investigation of the logic of theory change (alias belief change) [2]. AGM models are about the contraction of $\mathcal{L}_{Prop}$-theories, and focus on minimal change. In contrast, we here consider a modal language $\mathcal{L}_{\mathbf{Cxt}}$.[1] And our modal operator $-$ allows to express properties about a *family* of contractions, which actually do not necessarily satisfy the AGM criteria of minimal change. However, the validity $\neg[X]\psi \to (\varphi \leftrightarrow [X{-}\psi]\varphi)$ captures one of these minimality criteria. Another one is expressed by the valid formulas $\alpha \to [X{-}\psi][X{+}\psi]\alpha$ and $[Y]\alpha \to [X{-}\psi][X{+}\psi][Y]\alpha$, with $\alpha \in \mathcal{L}_{Prop}$, which correspond to the AGM principle of recovery. The invalid $\neg[X]p \to [X{-}p][X{+}p]\neg[X]p$ demonstrates that the above formula does not generalize to all α in $\mathcal{L}_{\mathbf{Cxt}}$.

[1] In fact, our formalism satisfies the same dynamic properties about Moore sentences as DEL [17].

Although formal analysis of norm change are available in the literature, the issue of a formal semantics for the dynamics of norms is relatively new. Indeed, most work in deontic logic is about defining formal semantics describing static deontic concepts. From this perspective, our research strategy is close in spirit to Segerberg's [13], who argued for an integration of AGM belief revision with Hintikka-like static logics of belief: we here do the same for deontic logic.

Among the few attempts to provide a formal semantics to norm change we here consider the approach proposed in [11]. There, an extension of the dynamic logic of permission (DLP) of [16] with operations of granting or revoking a permission was proposed. They call DLP$_{dyn}$ this DLP extension. Their operations are similar to our operations of norm promulgation and norm derogation. DLP is itself an extension of PDL (propositional dynamic logic) [10] where actions are used to label transitions from one state to another state in a model. The DLP$_{dyn}$ operation of granting a permission just augments the number of *permitted* transitions in a model, whereas the operation of revoking a permission reduces the number of *permitted* transitions. However there are important differences between our approach and Pucella & Weissman's. For us, normative systems are more basic than obligations and permissions, and the latter are defined from (and grounded on) the former. Moreover, dynamics of obligations and permissions are particular cases of normative system change (normative system expansion and contraction). Thus, we can safely argue that our approach is more general than Pucella & Weissman's in which only dynamics of permissions are considered. It is also to be noted that, while in our approach classificatory rules and their dynamics are crucial concepts in normative change, in DLP$_{dyn}$ they are not considered and even not expressible. In future work we will analyze the relationships between DLP$_{dyn}$ and our logic, and possibly a reduction of DLP$_{dyn}$ to our logic $\mathbf{DCxt}$.

While Pucella & Weissman's revocation of permissions corresponds to public announcements in DEL, no DEL approaches have proposed the counterpart of their operation of granting permissions, alias contractions (with the exception of [15], but in the framework of a logic of preference). Probably the reason for that is that it is difficult to define contraction operations both preserving standard properties of epistemic models such as transitivity and Euclidianity and allowing for reduction axioms. As we have shown, this is possible in our logic $\mathbf{DCxt}$ thanks to the intercontextual interaction axioms.

6 Conclusions

We have introduced a dynamic logic accounting for context change, and have analyzed several aspects of norm change, viz. the dynamics of permissions, obligations and classificatory rules. Although the logic has been applied here only to provide a formal analysis of norm-change, it is clear that its range of applications is much broader. Viewed in its generality, the logic is a logic of the dynamics of propositional theories, and as such, can be naturally applied to formal epistemology by studying theory-change, or to non-monotonic reasoning by studying how the context of an argumentation evolves during, for instance, a dialogue game. This kind of applications are future research. Another line of research would be to study the interaction between contexts, and so in a dynamic setting. Notice, in particular, that it would be straightforward to define a set algebra on contexts.

References

[1] T. Ågotnes, W. van der Hoek, J.A. Rodriguez-Aguilar, C. Sierra, and M. Wooldridge. On the logic of normative systems. In *Proc. of IJCAI'07*, pages 1181–1186. AAAI Press, 2007.

[2] C. Alchourrón, P. Gärdenfors, and D. Makinson. On the logic of theory change: Partial meet contraction and revision functions. *J. of Symbolic Logic*, 50:510–530, 1985.

[3] C. Alchourrón and D. Makinson. Hierarchies of regulations and their logic. In R. Hilpinen, editor, *Deontic Logic: Introductory and Systematic Readings*. D. Reidel, 1981.

[4] C. E. Alchourrón and E. Bulygin. *Normative Systems*. Springer-Verlag, 1971.

[5] A. Anderson. A reduction of deontic logic to alethic modal logic. *Mind*, 22:100–103, 1958.

[6] P. Blackburn, M. de Rijke, and Y. Venema. *Modal Logic*. Cambridge Univ. Press, 2001.

[7] C. Ghidini and F. Giunchiglia. Local models semantics, or contextual reasoning = locality + compatibility. *Artificial Intelligence*, 127(2):221–259, 2001.

[8] G. Governatori and A. Rotolo. Changing legal systems: abrogation and annulment (part I: revision of defeasible theories). In *Proc. of DEON'08*, LNAI, pages 3–18. Springer-Verlag, 2008.

[9] D. Grossi, J.-J.Ch. Meyer, and F. Dignum. The many faces of counts-as: A formal analysis of constitutive-rules. *J. of Applied Logic*, 6(2):192–217, 2008.

[10] D. Harel, D. Kozen, and J. Tiuryn. *Dynamic Logic*. MIT Press, Cambridge, 2000.

[11] R. Pucella and V Weissman. Reasoning about dynamic policies. In *Proc. FOSSACS'04*, LNCS, pages 453–467. Springer-Verlag, 2004.

[12] J. R. Searle. *Speech acts: An essay in the philosophy of language*. 1969.

[13] K. Segerberg. Two traditions in the logic of belief: bringing them together. In H. J. Ohlbach and U. Reyle, editors, *Logic, Language and Reasoning: essays in honour of Dov Gabbay*. Kluwer, 1999.

[14] R. Stalnaker. On the representation of context. *J. of Logic, Language, and Information*, 7:3–19, 1998.

[15] J. van Benthem and F. Liu. Dynamic logic of preference upgrade. *J. of Applied Non-Classical Logics*, 17(2):157–182, 2007.

[16] R. Van der Meyden. The dynamic logic of permission. *J. of Logic and Computation*, 6:465–479, 1996.

[17] H. van Ditmarsch, W. van der Hoek, and B. Kooi. *Dynamic Epistemic Logic*, volume 337 of *Synthese Library Series*. Springer, 2007.

Early requirements engineering for e-customs decision support: Assessing overlap in mental models

Brigitte Burgemeestre, Jianwei Liu, Joris Hulstijn, Yao-Hua Tan

Faculty of Economics and Business Administration,
Vrije Universiteit, Amsterdam,
{cburgemeestre, jliu, jhulstijn, ytan}@feweb.vu.nl

Abstract. Developing decision support systems is a complex process. It involves stakeholders with diverging interpretations of the task and domain. In this paper, we propose to use ontology mapping to make a detailed analysis of the overlaps and differences between mental models of stakeholders. The technique is applied to an extensive case study about EU customs regulations. Companies which can demonstrate to be 'in control' of the safety and security in the supply chain, may become 'Authorized Economic Operator' (AEO), and avoid inspections by customs. We focus on a decision support tool, AEO Digiscan, developed to assist companies with an AEO self-assessment. We compared the mental models of customs officials, with mental models of the developers of the tool. The results highlight important differences in the interpretation of the new regulations, which will lead to adaptations of the tool.

Keywords: e-government, shared mental models, decision support systems

1 Introduction

The creation, implementation and enforcement of legislation are complex processes that involve a large amount of people, parties and disciplines [8]. In this paper we discuss a decision support system to assist in such a complex regulatory environment. The European Union has drafted new customs legislation intended to make supply chains more secure. Trustworthy companies are certified by customs authorities to become 'Authorized Economic Operator' (AEO[1] [2]) and benefit from reduced customs inspections [1]. The AEO legislation has to be implemented by national customs, enforced by regional customs authorities and understood and applied by businesses. As a result, we observe the introduction of several decision support systems which try to support these tasks. To align the tasks of the stakeholders in the certification process, such decision support systems have to take complex stakeholder characteristics into account.

The phase of early requirements engineering aims to analyze stakeholder interests and how they might be addressed or compromised by system requirements [23] [5]. A well known approach to early requirements engineering is the *i** framework [23] which proposes an actor-oriented approach, based on the goals and intentions of an actor. It consists of two main modeling components: the Strategic Dependency (SD) model contains dependency relationships among actors in an organizational context, while the Strategic Rationale (SR) model describes stakeholder interests and

[1] http://www.douane.nl/zakelijk/aeo/en

[2] http://ec.europa.eu/taxation_customs/customs/policy_issues/customs_security

concerns, and how they are addressed by the system. An important issue that is not addressed by early requirements methods like *i**, is the existence of overlap or differences in the interpretations of the various stakeholders. Much work in requirements engineering implicitly assumes that mental models of the task and domain are shared among stakeholders. In practice however, this assumption is not always warranted. Overlap in task-specific knowledge structures or having a 'shared mental model' is argued to have a positive influence on performance and effectiveness in collaborative situations [8] [4] [14]. We argue therefore that early requirements engineering should involve identification of the differences and similarities that exists among the mental models of the stakeholders. With the differences clarified, the stakeholders become aware about each other's mental model constructs, which they in turn can use to align their approaches. Unlike some of the empirical work on shared mental models, however, we are not satisfied with mere lists of differences. Instead we propose to use conceptual models in the form of ontologies, in the sense of CommonKADS [17], as well as ontology mapping techniques, to detect divergent or synonymous concepts in two or more ontologies in a systematic and precise way.

The need to analyze mental models of stakeholders is particularly important in the development of innovative e-government solutions. E-government solutions aim to modernize and reorganize the public sector though new methods of governmental business [15]. Examples are one-stop government shops, public-private partnerships or outsourcing to the customer [20]. Especially in public-private partnerships, multiple parties are involved with different interests and backgrounds, leading to different interpretations. Moreover, the legislation involved in e-government solutions is often new or still evolving, which makes its interpretation also difficult for the regulator. This suggests that the regulator should be modeled like any other actor, with its own specific interests and beliefs about the task and domain. The idea to treat the regulator as any other actor is advocated by Boella et al [2]. Using Normative Multiagent Systems (NMAS), they analyze various regulative environments and study the interactive 'games' which agents play to determine whether it is in their interest to obey a norm or not, and for regulators, whether to enforce a norm or not, depending on the expected behavior of the other agents. For such games, the actual norms do not matter much; what matters are the perceptions agents have of the other agents' mental models of the norms.

In this paper we discuss the initial results of our research on assessing overlap in mental models. The research method is qualitative and empirical. We focus on a decision support system called 'AEO Digiscan' that supports companies in performing the self-assessment, which is required to obtain AEO certification. We have conducted interviews with experts from both the Dutch Customs and Tax Administration (DTCA) and from the consultancy firm Deloitte, who have developed the tool and who are using it to assist their clients in the AEO certification process. We compare the interview results to identify differences in the expert interpretation of the AEO self-assessment task and in the requirements to obtain AEO certification. To structure the analysis of the expert interpretations, we use conceptual models taken from the CommonKADS methodology [17] and from the literature on risk management. Overlaps and differences between interpretations are mapped, using ontology mapping [18] [12].

The remainder of the paper is organized as follows: Section 2 describes our approach towards a conceptual model of mental model mappings; Section 3 describes our analysis of the case study of AEO self-assessment. The paper ends with a discussion and conclusion of our results.

2 Towards a conceptual model

To identify requirements for an innovative E-government solution that concerns public-private partnerships, such as the AEO certification procedure, we propose Normative Multiagent Systems (NMAS) as a starting point for an analysis. Each stakeholder is viewed as an autonomous agent that can act, perceive its environment, communicate with others and has skills to achieve its goals and tendencies [22]. Although agents are autonomous, their behavior must be restricted by norms. The regulator, which enforces the norms, is also seen as one of the agents and not as a separate entity [2]. This makes sense in our case, because for public-private partnerships, both regulator and businesses have to interpret the legislation to apply it in practice. Figure 1 shows a situation in which two agents 'A' and 'B' must collaborate. To do so, they must interpret norms, and implement them in practice. For each agent we draw two 'thinking balloons': the agent's own interpretation of the norms, and the agent's beliefs about the other agent's interpretation of the norms.

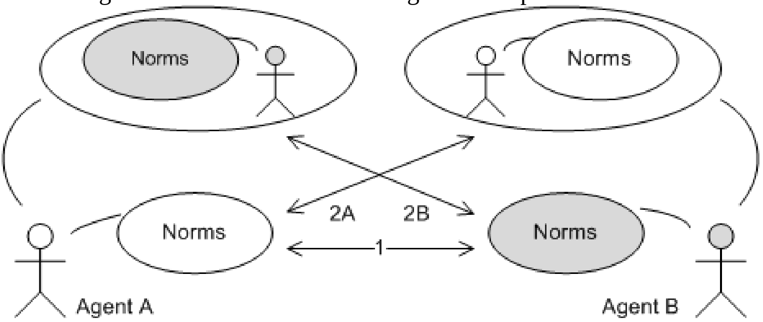


Fig. 1. Agents' beliefs about the norms, and about each other's beliefs of the norms

We suggest that for successful collaboration both agents must have either a shared interpretation of the norms or that their mental models are transparent for the other, so that other agents can adjust their behavior and overcome differences. Uschold and Gruninger [20] also argue that for software agents or IT systems to successfully communicate with each other, they need to be semantically integrated. Successful exchange of information means that agents understand each other and accuracy is guaranteed [20]. This requires that agents must agree on a communication standard or protocol and a common ontology. However in the real world often various ontologies exist about a single topic, so we better speak of semantic heterogeneity than of semantic interoperability [12] .We therefore include a need for transparency in our model. The assumption is that if agents have knowledge about each others' interpretation of the norms, they can predict each others' behavior and task performance, and can adjust their actions accordingly.

To analyze the expected effectiveness of the collaboration we can therefore compare the thinking balloons in two ways (see Figure 1): arrow 1 compares the agent's mental models of the norms, and arrow 2A en 2B compare the mental model with the beliefs the other agent has about the mental model. We also note that the agent's mental model and the belief about the other agent's model can influence each other but this interaction is not addressed in this paper. To assess overlap between the mental models and the beliefs about the mental models we use a technique from software engineering: ontology mapping [18] [12]. Ontology mapping techniques and formalisms are intended to overcome the issue of heterogeneity by identifying the differences and similarities between ontologies. We view the agents in our example as two agents that need to have a (partial) mapping of their ontologies to communicate and collaborate effectively. To promote the merger of ontologies towards semantically interoperable ontologies a first step is to identify the overlapping concepts and key differences. With the differences and commonalities made explicit, the agents become aware about each others mental models, which can in turn help them to more effectively discuss and overcome the differences.

There are various techniques for finding correspondences between semantically related entities of different ontologies. Most matching techniques require the existence of a commonly shared body of knowledge, structure, language or syntax. However in an innovative public private partnership where both the businesses and government have to adapt to their new roles, commonly known responsibilities and ways of interaction do not exist yet. The shared body of knowledge is evolving as best practices are developed, procedures are maturing and lessons are learned based on experiences in the field. Research with a multi agent systems viewpoint does address this issue with the introduction of meaning negotiation or semantic negotiation [3] [6]. These techniques offer a dynamic and flexible form of semantic coordination for situations in which no a priori coordination exists. Bouquet et al. introduce in [3] a method that makes the meaning of nodes in structured semantic models explicit by combining three types of knowledge: lexical, domain and structural knowledge. They combine the knowledge sources to build a new representation of the problem, where the meaning is encoded as a set of logical formulae. Another approach to match ontologies is provided by instance based methods [**7**] [16]. These methods focus on the most active parts of the ontologies and reflect the semantics of the concepts as they are actually being used [16]. Instance-based ontology matching techniques determine the similarity between concepts of different ontologies by examining the extensional information of concepts [**7**]. Various approaches to instance based methods exist: in [**7**] machine learning techniques are used to identify mappings and in [16] a lexical search engine is used to map instances from different ontologies. Concept classification information is exchanged between these mapped instances, to generate an artificial set of common instances shared by concepts from two ontologies, so that simple similarity measures can be applied. The advantages of this method are that it does not depend on the availability of concept labels or a rich ontology structure.

For the matching of mental models of agents in a regulatory setting in which no prior coordination model exist we propose a combination of techniques and different knowledge sources. To construct the mental models in a structured way and to function as a common reference model we propose the use of generic knowledge model templates, from knowledge engineering methods such as CommonKADS [17]. The domain independent nature of such templates provides a good basis for the agent specific models. In line with the CommonKADS method, the agent's models we construct will therefore consist of three knowledge categories: domain knowledge, task knowledge and inference knowledge [17]. Besides that we use legislation and norm frameworks as background domain knowledge to assess the validity of mappings. Then we can determine if concepts relate to the same topic and have a similar or compatible meaning. Furthermore we use instances, implementations of the norms, to derive concepts and mappings. We illustrate the method by a short example of different interpretations of risk assessment, taken from the case study.

The 'Assessment' knowledge model template [17] will function as a starting point to model the risk assessment approaches of a company and the regulator. We can then compare the deviations of the approaches with the original model and since the skeleton is similar we can also compare both risk assessment approaches. To assess the validity of matched concepts we use legislation to determine the meaning. For example the concept *security* can be aimed at preventing theft, taking goods out of the supply chain or preventing smuggling and terrorism, adding things to the supply chain, or a combination of both. Furthermore we use observations in the real world to trace back to which concepts they refer. For example a gate can be seen as an instance of the concept *measure* to prevent intruders from entering a company's premises. While a personal policy can also be seen as a *measure* to avoid the hiring of untrustworthy personnel.

Combining these issues, we come to a three step approach to analyze and compare mental models of agents. Step 1 is to develop generic domain, task and inference models based on knowledge templates from CommonKADS [17]. These generic models are used as a starting point for constructing the agent's specific mental models. Step 2 is to use the generic models to externalize, analyze and compare individual agent's mental model constructs. Step 3 is to build a conceptual model that presents the encountered differences and similarities of the mental models of the agents. This model makes the differences in mental models transparent, which makes it easier to overcome the heterogeneity or to adjust the models accordingly. The following section describes the application of this approach to a case study.

3 Case study: AEO self-assessment of a petrochemical company

We use the approach described in the previous section to analyze a specific case of an AEO self-assessment, which is part of the application procedure for companies to qualify for AEO. The AEO self-assessment is a nice example of collaboration between public and private parties, because a traditionally public task (AEO assessment) is partly delegated to a private party (a company). The private party therefore needs insight in the mental model of the public party (customs authority) to perform the task according to their standards. The customs, on the other hand, are interested in the mental model of the company, because the legislation is new and customs need to learn from best practices of early AEO applicants. The next paragraph provides a short introduction to the AEO legislation and certification procedure.

3.1 AEO legislation and certification

An Authorized Economic Operator (AEO) can be defined as a company that is reliable throughout the EU in the context of its customs related operations [9] [10] [12]. The holder of an AEO certificate will receive several benefits in customs handling within all EU member states that can lead to considerable cost-reductions for businesses. The degree to which a company is granted these facilities depends on the type of certificate: ‘Customs simplifications’, ‘Security and safety’ or ‘Combined’. For non-certified enterprises customs will continue to carry out the traditional supervision. The flow of goods for customs will therefore consist of two parts: goods from AEOs and goods from non-certified companies. Customs can direct their efforts towards non-certified companies to increase the security of international supply chains, while at the same time reducing the administrative burden for AEOs.

To qualify for the AEO status a company must meet a number of criteria, which are described in the community customs code and the AEO guidelines [9].The general customs’ certification practice is that customs officials visit a company which applied for a license, to assess whether the company complies with the legislation and whether a license can be issued. In the AEO certification procedure however, a company must first perform a self-assessment of their compliance to the AEO legislation. The left swim lane in Figure 2 presents the steps that a company has to perform in the self-assessment and the right swim lane shows the activities of the customs in the AEO certification. The first step is that a company collects information relevant for the AEO status, such as business processes, safety procedures, licenses and certificates, IT systems, etc. The next steps are to identify the (potential) risks to which the business is exposed (using the AEO guidelines), to identify the measures that are implemented to mitigate these risks, and to further specify the generic AEO criteria and turn them into internal norms which evaluate the risk mitigation in relation to the line of business. For example, computer components are valuable goods, which are subject to theft. Trading valuable goods requires more security measures, than, say, trading in a mass product like fertilizer. However, some ingredients of fertilizer may be used to assemble explosives, leading to a different set of risks. By evaluating the risk mitigation strategies, a company must determine if the risks are mitigated sufficiently, or if additional measures are needed.Then a company must evaluate the effective implementation of the proposed measures, using the COSO internal control scoring definitions, which are part of the summary of the AEO self-assessment. The scores range from 0 “no control measures in place” until 5 “internal control measures are integrated into the business processes and continuously evaluated”. After that the company either submits the AEO application or implements (additional) measures.

Once the customs receive the AEO application, they assess whether it is a valid application according to entry conditions. Next, they determine the type of visit, based on the AEO application and on historical data about customs and tax compliance. A visit is needed to check whether the self-assessment is performed correctly and whether the company identified all the risks and has taken all appropriate measures.

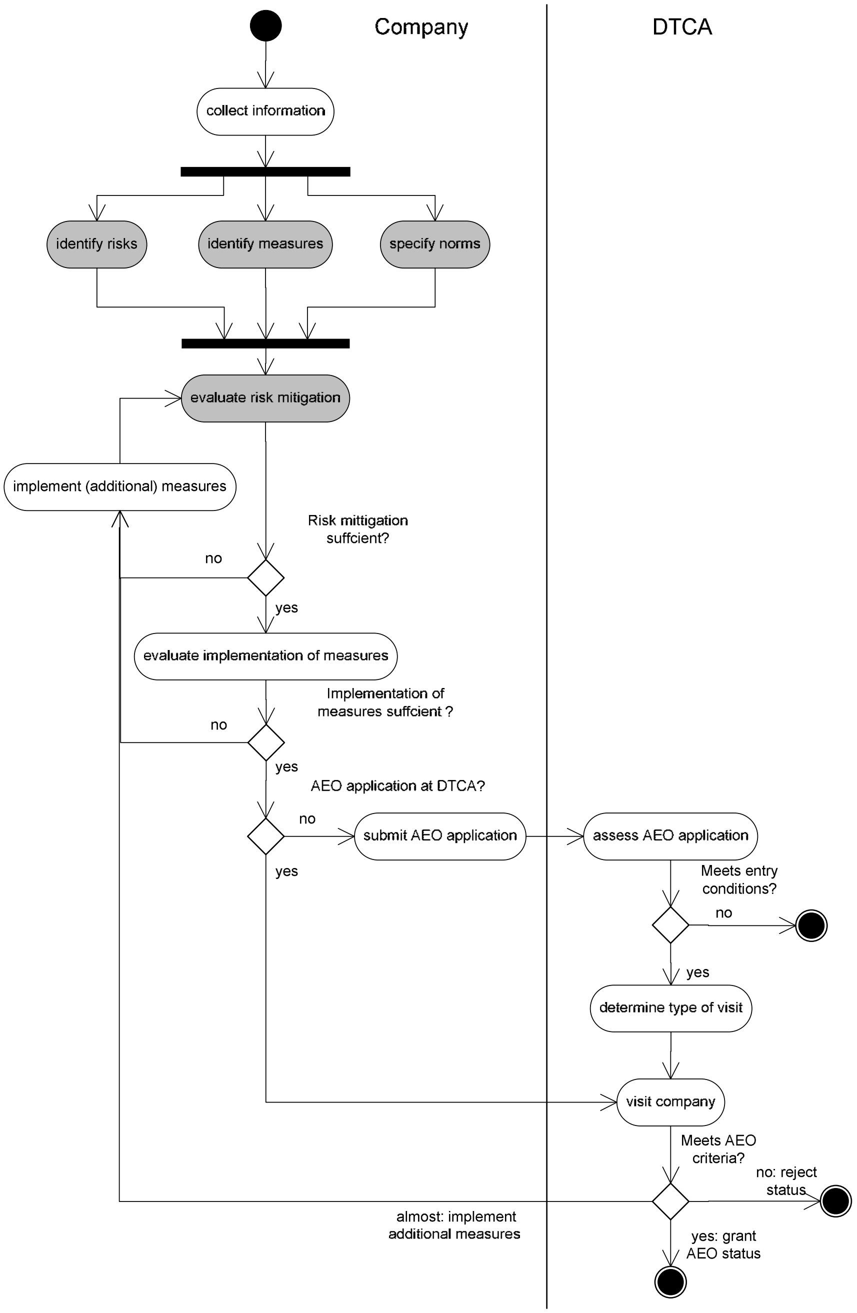


Fig. 2. Activity diagram for the AEO certification procedure. Activities in grey are supported by the AEO Digiscan.

Based on the visit, customs determine whether the AEO certificate is granted or not or that first additional measures need to be implemented. In that case, customs will visit the company a second time, to check if the additional measures are implemented.

Ideally, a company would perform the self-assessment like customs would, when they are ‘auditing’ a company for AEO compliance. The customs authority could then rely on the findings of the company and minimize their own visit. However, from interviews with DTCA officials we learned that companies often find it difficult to perform an AEO self-assessment. Consultancy firms therefore offer services and tools to assist companies. One of these firms is Deloitte and their tool is called the AEO Digiscan. The steps in the process which are supported by the Digiscan are colored dark in Figure 2. The next section describes the Deloitte AEO Digiscan in more detail.

3.2 AEO Digiscan

To support companies in performing the AEO self-assessment Deloitte’s Tax Advise unit developed the AEO Digiscan. The AEO Digiscan is an online tool that works as a classic expert system. It contains rules, which represent the AEO guidelines and the sections in the questionnaire are organized accordingly. Various experts of Deloitte such as tax advisors, security specialists, IT specialists and auditors contributed to the development of the AEO Digiscan, by specifying the guidelines, and turning them into clear questions. The questions that a company has to answer depend on the company’s role in the supply chain and on answers to earlier questions. Scores are expressed on a 5 point scale ranging from red (1) till green (5). For example, red (1) means “Potential risk can be considered high”, orange (3) means “Potential risk could neither be considered low nor high” and green (5) means “Potential risk could be considered low and acceptable”. The score of each section is based on the lowest score in the section and cannot be altered after a section is completed. After answering the questions, experts of Deloitte check the AEO Digiscan results. They have the possibility to adjust the scoring if they think a company has overestimated or underestimated its record. After that the risk based score of the AEO Digiscan is automatically translated into the COSO based scoring used by DTCA, and the AEO summary is filled out. Deloitte sends the AEO Digiscan report and the AEO summary with feedback to the company. The company can then decide to send the AEO application to DTCA.

The added value of the tool is that it provides a structured approach to AEO self-assessment. It assists companies in interpreting and applying the AEO guidelines. Furthermore, it provides companies with an indication of their position with respect to achieving the AEO status, and points out their strengths and weaknesses.

When a company uses the AEO Digiscan to perform a self-assessment, we can view this as another delegation of the self-assessment task, namely to Deloitte‘s AEO Digiscan. To assure that the self-assessment task is performed as intended by DTCA, it is therefore important to also assess the overlap between the mental models of Deloitte and DTCA, besides the regular mapping between company and DTCA. In this paper we will focus on the mapping between Deloitte and DTCA. The next section describes our analysis of the AEO self-assessment task.

3.3 Case analysis

This section presents our analysis of the differences and overlap that exist between the approaches of AEO self-assessment of DTCA and Deloitte. We perform our analysis according to the steps described in Section 2. For the data collection we used the following methods: document analysis and semi-structured interviews [24]. We studied internal and public documents from both DTCA and Deloitte that describe their vision and approach on AEO certification and self-assessment. To gain insight in the expert interpretation of the AEO self-assessment, we conducted 5 interviews with both DTCA and Deloitte, held one meeting were we invited both parties simultaneously, joined DTCA auditors on their first visit to a petrochemical company and held a first feedback session for both DTCA and Deloitte to present our initial research results. To elicit detailed expert knowledge, we showed the experts the AEO application of a petrochemical company “PCC”, which had used the Deloitte AEO Digiscan, and asked them how they would have assessed this company (if there would have been no AEO self-assessment) and if they could point out points of interest. We also asked them some questions about the AEO certification and self-assessment in general. In total we have spoken with 10 persons from DTCA and 5 from Deloitte. The duration of the interviews varied from 2- 4 hours. Except for the visit, the meeting and a first interview with Deloitte, we tape-recorded all interviews with the participants’ prior agreement. Minutes were made of meetings.

3.3.2 Domain, task and inference model

To analyze the interview results, we use an adapted version of the knowledge model templates for the assessment task of the CommonKADS methodology [17].To save space; we do not show a task model in this paper. Figure 3 represents the domain schema for AEO certification. The purpose of this model is to specify key concepts and indicate how they are related. The implementation of these relationships is then further worked out in the inference structure, which we present in Figure 4.

First we have to identify the domain. A company is eligible for an AEO certificate, when it conforms to four criteria: (1) an appropriate record of compliance with customs regulations, (2) sufficient internal control measures regarding trading and logistics, to allow for customs auditing, (3) conformance with certain solvability criteria, and (4) appropriate security measures to safeguard the supply chain. In the interpretation of DTCA, the AEO self-assessment is essentially a statement in which the company declares to be `in control’ of its supply chain. Under the current interpretation of DTCA, this means that the company must have performed a risk assessment to identify key risks regarding security in the supply chain, must have taken appropriate control measures to mitigate the risks, and must have evidence that these measures have been operationally effective. So a conceptual model of risk management seems a good starting point for domain analysis. Risk management is the activity – performed by management – of continuously assessing risks, defining and implementing control measures to mitigate risks and evaluating and improving the results. A well known best practice for IT risk management has been proposed by NIST. They define a risk as a function of the likelihood of a given threat-source exercising a particular potential vulnerability, and the resulting impact of that adverse event on the organization [19]. Similar definitions are found in other literature on risk management.

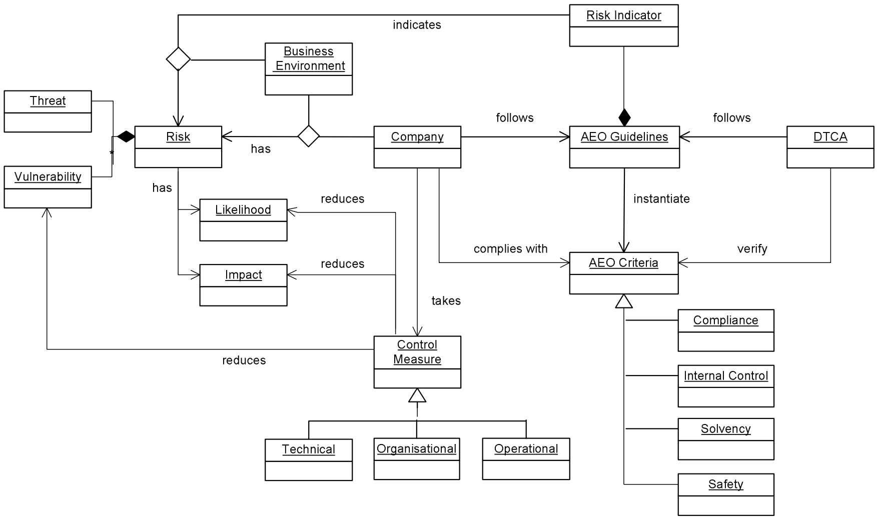


Fig. 3. Domain schema for AEO certification

The left of Figure 3 shows general risk assessment concepts. A risk assessment identifies the threats facing a company given its line of business and its environment. The vulnerability of a company to threats depends on its current control measures. Control measures either reduce the likelihood, by dealing with vulnerabilities (preventative controls), or reduce the impact (detective and corrective controls). Consider for example the risk of smuggling: someone secretly places an additional item in a container. This vulnerability can be reduced by limiting physical access to all premises where containers are loaded and unloaded, to those employees who need to have access because of their job. In general, there are three kinds of control measures: technical controls (e.g. authentication by RFID badge), organizational controls (e.g. access control based on real needs) and operational controls (e.g. reconciliation of shipping order against inventory). On the right we show the AEO criteria and the AEO guidelines. The guidelines do not act like norms, as one might expect. They are merely high-level attention points, which – given a business environment – indicate the main risks for the company. It is the responsibility of the company to set their own internal norms, depending on the actual risks encountered.

Figure 4 depicts the inference structure for AEO self-assessment. It is the generic assessment model, taken from [17]. The input for the inference is the case, a description of the company that applies for AEO status. First, a company must abstract case data that corresponds with the data used in the norms. For the AEO self-assessment this means that a company has to identify all the potential risks, the measures that mitigate these risks, and the implementation of the measures, related to its business activities and role in the supply chain. A company must then specify which (sub) sections addressed in the AEO guidelines are applicable to the company's specific situation and need to be evaluated and reported in the AEO summary. From this set of (sub) sections a company selects a single

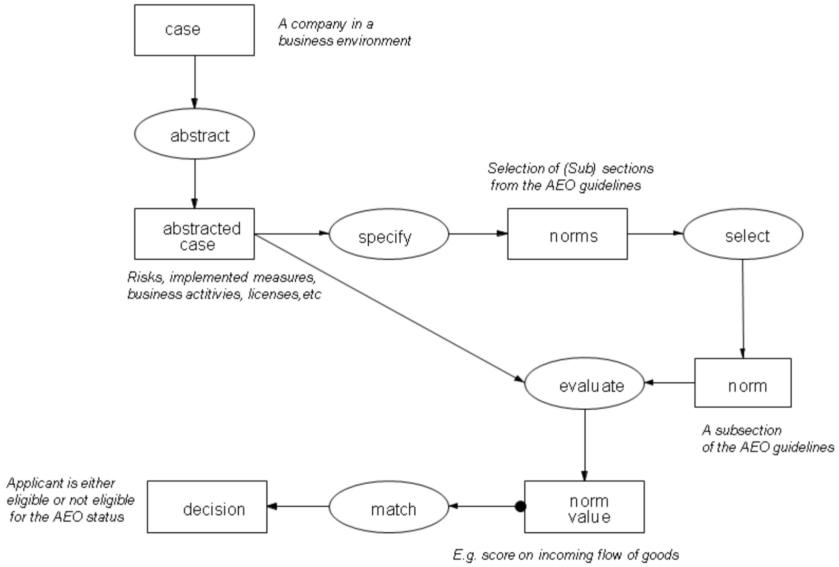

case
A company in a business environment
abstract
Selection of (Sub) sections from the AEO guidelines
abstracted case
specify
norms
select
Risks, implemented measures, business actitivies, licenses,etc
evaluate
norm
A subsection of the AEO guidelines
Applicant is either eligible or not eligible for the AEO status
decision
match
norm value
E.g. score on incoming flow of goods

Fig. 4. Inference structure of the assessment task (Schreiber et al 2000)

subsection for evaluation. For each subsection a company determines if the risk mitigation is sufficient and evaluates the implementation of the measures. The output value is an integer (0-5) indicating the implementation level of the measures, which a company reports in the AEO summary. The match function checks whether the scores on the self-assessment summary lead to a decision if a company is AEO compliant or not. The match function only stops prematurely in case of (clear) incompliance. A company is only AEO compliant when it scores well on all the (sub) sections that are applicable.

3.3.3 Constructing and comparing mental models

Now we present the interview findings, organized according to the inference model of the previous paragraph.

Abstract: The 'abstraction' inference is a complex step. Essentially it is a form of classification, which abstracts over individual differences. According to DTCA, to properly evaluate the mitigation of risks, they have to be evaluated in context. This includes the business activities, company role in the supply chain, organizational structure, location, etc. Case data about all these aspects and their interaction, needs to be combined in an abstract classification. There is no structured approach available to classify the type of company; DTCA only advises the companies to use the AEO guidelines to identify risks. Classification in the AEO Digiscan is a lot simpler. It only looks at risks and measures related to the company's role in the supply chain.

Specify: The AEO guidelines contain a table that indicates which of the (sub) sections of the guidelines are applicable, based on the company's role in the supply chain. A company can also decide to leave out or include certain subsections based on its specific business activities, e.g. when a company is both a manufacturer and an exporter. The AEO Digiscan makes use of this table and automatically presents only the questions related to the company's role in the supply chain. Experts of Deloitte made the AEO guidelines more specific, specifying questions that are easy to understand. Based on previous answers the tool selects the next question. However, DTCA officials believe that the AEO guidelines have been implemented too literally and that it does not take the business environment into account. For example, the AEO Digiscan contains very general questions about IT, such as: "Which operation system is used in your company?" to which PCC answered: "Windows". However, no questions are asked about the IT systems used in the manufacturing process. PCC is partially a manufacturer, so a risk to its key business processes is a threat to a secure supply chain. DTCA officials also realize the limitations of a tool like this, and wonder how an electronic questionnaire can ever be complete, if it has to take all these specific characteristics into account.

Select: The DTCA approach requires the manual selection of subsections of the guidelines. The AEO Digiscan automatically selects and presents a question, on the basis of answers to the previous questions.

Evaluate: The evaluation step requires companies to first perform a risk assessment, in which the adequacy of the risk mitigation is assessed relative to the business context, and second to evaluate the implementation of these measures. DTCA does not provide a step by step approach to do the evaluation. A company must itself determine its COSO level on all applicable subsections in the AEO summary. The AEO Digiscan focuses on risk assessment and identifies potential risks and the measures that are in place. After a section is completed, the tool automatically calculates the potential risk level for the subsections and the whole section. According to a DTCA official: *"A tool should not let people answer questions without knowing why they answer them. It should first give a good overview of the purpose of the specific questions"*. If people do not understand the purpose of a question, they can misinterpret the question and give the wrong answer. Furthermore, hiding the 'abstraction' inference from the user turns the self-assessment into a checklist that can be filled out without creating awareness on internal control or safety measures.

Match: After the AEO Digiscan is completed, it provides for each subsection an indication of the company's position with respect to achieving the AEO status. To prevent fraud, DTCA does not tell companies what a sufficient score is to achieve the AEO status. The companies receive the first feedback on their scores during the customs visit.

In general we find that the approach offered by the AEO Digiscan is more structured and requires less expertise on AEO legislation, than the general approach that is proposed by DTCA. However, the scope of the AEO Digiscan is limited; it focuses on risk assessment (identifying risks and measures) while DTCA's approach focuses on risk management, including implementation of measures. Although the tool is limited, it provides for a consistent assessment process. DTCA officials asked for insight in the scoring calculation mechanism of the AEO Digiscan. Deloitte would have liked more insight in DTCA's requirements and into their evaluation approach. Furthermore we noticed that DTCA pays a lot of attention to the reliability of the self-assessment and to the way it was performed, while Deloitte's focus is more on specifying the AEO legislation and AEO guidelines.

3.3.4 A conceptual model of scoring

We will further zoom in on the differences in the scoring model, which is an important issue according to both parties. The grey concepts in Figure 5 are only covered by the DTCA approach; the white concepts are part of both approaches. We observe that the AEO Digiscan covers only part of the DTCA approach. The AEO Digiscan focuses on risk assessment, whereas the self-assessment, as it is interpreted by customs, involves risk management, which also stresses the need for additional measures and evaluation. This is in line with the views that DTCA and Deloitte have on AEO certification. DTCA sees the AEO self-assessment as a means to judge the quality of companies' internal control system, and to create awareness of potential risks. In contrast, Deloitte efficiently provides companies with an indication of their readiness to achieve AEO status. The Deloitte approach is therefore more aimed at compliance with AEO legislation, whereas the DTCA approach aims at companies being 'in control' of their internal procedures regarding safety and security. The AEO Digiscan tool supports the compliance assessment through a bottom up approach: answer specific questions to arrive at an overall score. DTCA's approach works top down: to be in control, what measures does a company need to have implemented?

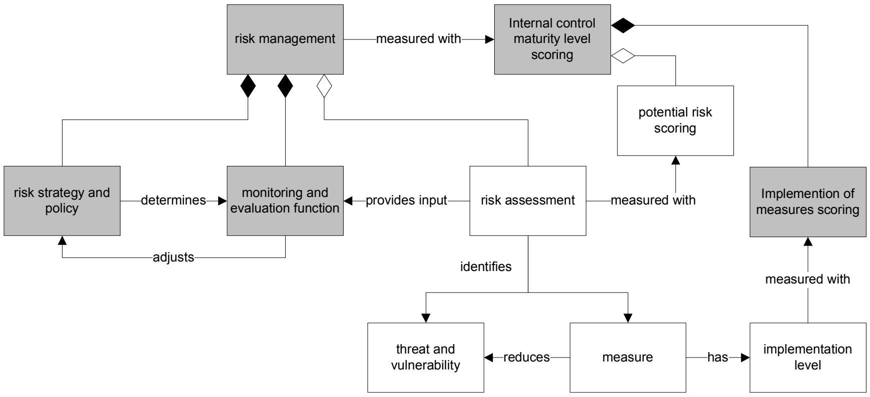


Fig. 5. Model of differences (dark) and overlap between DTCA and Deloitte

The different scoring models are in line with these different views on self-assessment. DTCA uses the COSO scoring, which measures the implementation of control measures and Deloitte uses a risk-based scoring. By making the differences in the scoring models explicit we pointed out to DTCA and Deloitte that Deloitte's risk-based approach is a step within DTCA's approach rather than a complete different approach. Aligning the approaches is therefore easier to achieve than it looked at face value. The interpretation of all these aspects needs to be addressed in the early requirements phase as they can lead to various system requirements. Should the AEO Digiscan support DTCA's risk management approach or should Deloitte focus on risk assessment only, and embed its tool in DTCA's approach? Should we use a risk based scoring method and do we need to include the implementation of the measures? This greatly influences the kind of tool that is developed and the role the tool will fulfill within the task of "self-assessment".

4. Discussion and conclusions

4.1 Regarding the research method

This paper reports on initial exploratory research. Interviewing proved to be a good research technique to gain insight into the AEO self-assessment approach of both Deloitte and DTCA, as our interviews uncovered some very interesting issues. However the number of interviews was limited, especially for Deloitte, where we only interviewed 5 people. Furthermore, our interviews were semi-structured and therefore not all topics were addressed consistently in all interviews. We therefore want to validate these results with a second round of interviews, using a more controlled set up. Another point is that we compared the expert knowledge embedded in a tool with real expert knowledge. The embedded knowledge was more explicit and therefore easier to compare, but it is already a selection of the expert knowledge of the Deloitte experts. On the other hand the AEO Digiscan gave us a good view on which part of the expert knowledge is easy to externalize and to imbed in a tool. Besides that we ourselves made the task and domain models based on the interview findings. It can therefore be argued that another interpretation was added to the mental models of the experts. In fact, we also compared original models used by the experts. However, since the Deloitte approach is based on the DTCA approach, comparing the models did not provide any results. The differences we encountered in the interviews were more concerned with the interpretation of the concepts by the experts. The expert interpretation of the domain (see Figure 3) was shared among the experts of both parties.

4.2 Regarding the mapping of mental models

Based on our interview findings we can conclude that by and large, the interpretations of the task and domain model for AEO self-assessment by experts from Deloitte and from DTCA overlap. Both make use of risk analysis methods and are based on the AEO guidelines, and therefore use similar attention points. However, important aspects of the self-assessment are interpreted differently. Regarding the task, there is disagreement about the scope of the self-assessment: does it only contain risk assessment (AEO Digiscan) or is it concerned with risk management, which also includes the implementation and constant evaluation of control measures (DTCA)? These task differences also show up in the domain analysis and the inference scheme. In particular, they lead to different scoring models: a risk-based model for AEO Digiscan, and COSO-based maturity levels for DTCA. There are also diverging ideas about the role of ‘understanding the business’ when assessing risks and controls. DTCA experts stress that control measures must be understood in context. For example, the strength of password protection must be interpreted relative to the business environment and IT infrastructure. Deloitte experts, on the other hand, have tried to further specify and instantiate the generic AEO guidelines into specific questions. Moreover, an initial classification of the company will automatically select only the relevant questions. But despite such customization, the tool does not allow for any company specific considerations. As a benefit, this generic nature of the AEO Digiscan improves the transparency and reliability of the assessment procedure. Our methodology does not require completely shared mental models. Differences of opinion or mental model are fine, as long as parties know the differences, and know how to adjust their behavior accordingly (Figure 1). For requirements engineering, this means that mental models about other stakeholders have to be modeled explicitly as this helps them to realize what their respective positions are, and act accordingly. From the case study it even became clear that making differences explicit is often the first step towards solving the differences. Our analysis made the Deloitte experts aware what the differences between both approaches exactly are and that the difference concerned a scope problem rather than a complete mismatch. This insight has led to Deloitte taking action to adapt their tool and risk-based scoring model, to increase the overlap between their and DTCA's approach. In contrast with some of the empirical literature on shared mental models [4][14]we have attempted to make mappings of the actual differences and overlaps. To this end, we have used template models from CommonKADS [17]. Regarding these templates we can conclude that they have been instrumental in bringing out and explaining some key differences. For example, the difference between a case description and an abstracted case (Figure 4) turns out to reflect the effects of the loss of information in the AEO summary. Also the activities of specifying and selecting norms (Figure 4) explain important differences of opinion.

4.3 Regarding the AEO Digiscan decision support tool

Charting the differences between mental models of stakeholders is an important element of developing a complex decision support system, because it helps to identify differences in expected functionality, and in the way the system is expected to be used. Differences in task and domain models will lead to different system requirements, consider for example the scoring models. Therefore such mental model mapping should be part of early requirements engineering [5][23]. Note that some expectations may be too complex. It is easier to design and implement an expert system about compliance (rule-based), than about risk assessment in context (principle-based). A less ambitious system, with a task that naturally aligns with one or more sub-tasks of the task model, may be easier to get accepted, than an overly ambitious system which will disappoint some stakeholders.

Mapping overlaps and differences is especially important in a regulatory context. The regulator is leading. But also the regulator needs material on which to base its benchmarking. It cannot develop norms by itself, but has to use 'best practices' of companies. The experience of using a decision support tool has proved very useful in this respect, as the tool has forced experts to be specific about their intentions.

For future research we would like to narrow the focus of the research and try to make a more elaborate analysis of the differences. We are currently arranging more interviews with IT auditors from both Deloitte and DTCA, to zoom in on the IT aspects of AEO certification.

Acknowledgments. This research is part of the integrated project ITIADE which is funded by the 6th Framework IST Programme of the European Commission. We are grateful for the open discussions with representatives of DTCA, Deloitte and PCC.

References

1. Baida, Z., Rukanova,B., Liu,J. & Tan,Y. (2008)Preserving Control in Trade Procedure Redesign - The Beer Living Lab,Electronic Markets, The International Journal, Vol. 18, No. 1, pages53-64
2. Boella, G. and van der Torre, L. (2007). Norm negotiation in multiagent systems. International Journal of Cooperative Information 16(2), pp. 97-122.
3. Bouquet, P. Serafini, L. and Zanobini, S. Semantic Coordination: A New Approach and an Application, Proc. ISWC2003, Springer, LNCS 2870, 130-145, 2003.
4. Cannon-Bowers, J.A. & Salas, E. (2001) Reflections on Shared Cognition. Journal of Organizational Behavior, Vol. 22, No. 2, pp. 195-202
5. Castro, J., Kolp, M., and Mylopoulos, M. (2002) Towards requirements-driven information systems engineering: The Tropos project, Information Systems (27), pp. 365–389.
6. Van Diggelen, J., Beun, R. , Dignum,F., van Eijk, R., Meyer, J. (2007) Ontology Negotiation: Goals, Requirements and Implementation. International Journal of Agent-Oriented Software Engineering 1(1):63–90.
7. Doan, A.H., Madhavan, J., Domingos, P., Halevy, A. (2002) Learning to map between ontologies on the semantic web. In: Proceedings of the 11th international conference on World Wide Web, pp. 662–673
8. van Engers, T.M., Kordelaar, P.J.M., den Hartog, J., Glassée, E., (2000) POWER: Programme for an Ontology based Working Environment for modeling and use of Regulations and legislation. Proceedings 11th workshop on Database and Expert Systems Applications (IEEE) Greenwich London, pp. 327-334.
9. European Commission (2007) AEO Guidelines, TAXUD/2006/1450.
10. European Commission (2006) The AEO Compact model, TAXUD/2006/1452.
11. Euzenat, J., Shvaiko, P. Ontology Matching. Springer, Heidelberg (2007)
12. Kalfoglou, Y. & Schorlemmer,M.(2004) Formal support for representing and automating semantic interoperability. ESWS 2004, pp. 45–60
13. Kalfoglou, Y. & Schorlemmer, M. (2003) Ontology mapping: The state of the art. The Knowledge Engineering Review, Vol. 18:1, 1–31
14. Mohammed, S. & Dumville, B. C. (2001). Team Mental Models in a Team Knowledge Framework: Expanding Theory and Measurement Across Disciplinary Boundaries. Journal of Organizational Behavior, 22(March): 89-106.
15. OECD (2005). E-Government for Better Government, Organization for Economic Co-operation and Development.
16. Schopman, B.A. C. Wang, S. and Schlobach, S.(2002) Deriving Concept Mappings through Instance Mappings, In John Domingue and Chutiporn Anutariya, editors, ASWC, volume 5367 of Lecture Notes in Computer Science, pages 122-136
17. Schreiber, G., Akkermans, H., Anjewierden, A. , de Hoog, R., Shadbolt, N., Van de Velde, W. and Wielinga, B. (2000) Knowledge engineering and management, MIT Press.
18. Sowa, J. (2000) Knowledge Representation: Logical, Philosophical, and Computational Foundations. MIT Press.

19. Stoneburger, G., Goguen, A. and Feringa, A. (2005) Risk Management Guide for Information Technology Systems. NIST Special Publication 800-30
20. Uschold, M. & Michael Gruninger, M. (2004) Creating Semantically Integrated Communities on the World Wide Web. Invited Talk Semantic Web Workshop
21. Wimmer, M.A. (2002) A European perspective towards online one-stop government: the eGOV project. Electronic Commerce Research and Applications 1(1): 92-103
22. Wooldridge, M. (2002) An Introduction to Multiagent Systems, John Wiley & Sons (Chichester, England).
23. Yu, E.K.S. (1997) Towards Modelling and Reasoning Support for Early-Phase Requirements Engineering, in: Proceedings of the Third IEEE International Symposium on Requirements engineering, pp. 226-235.
24. Yin, R. K. (2003) Case study research: Design and methods. Sage Publications Inc.

FSL — Fibred Security Language

Valerio Genovese[1], Dov M. Gabbay[2], Guido Boella[1], Leon van der Torre[3]

[1] Dipartimento di Informatica. Università di Torino - IT.
E-mail: guido@di.unito.it; valerio.click@gmail.com
[2] Dept. Computer Science, King's College London - UK.
E-mail: dov.gabbay@kcl.ac.uk
[3] Computer Science and Communications, University of Luxembourg, Luxembourg.
E-mail: leon.vandertorre@uni.lu

Abstract. We develop a fibred security language capable to express statements of the form

$$\{x\}\varphi(x) \textbf{ says } \psi$$

where $\{x\}\varphi(x)$ is the set of all x that satisfy φ and ψ is any formula. φ and ψ may share several free variables. For example, we can express the following: "A member m of the Program Committee can not accept a paper P_1 in which one of its authors says that he has published a paper with him after 2007"

$$\neg(\{m\}[PC(m) \wedge \{y\}author\_of(y, P_1) \textbf{ says } \exists p(paper(p) \wedge author\_of(m, p) \wedge author\_of(y, p) \wedge year(p) \geq 2007)] \textbf{ says } accept(P_1))$$

1 Introduction

Access control is a pervasive issue in security: it consists in determining whether the principal (a machine, user, program) that issues a request to access a resource should be trusted on its request, i.e., if it is authorized. Authorization can be based in the simplest case on access control lists (ACL) associated with resources or with capabilities held by principals, but it may be complicated by, for instance, membership of groups, roles and delegation. Thus, logics for access control are often used to express policies and to enable reasoning about principals and their requests, and other general statements.

In many cases first-order/propositional logic suffices, but it does not in the case of distributed policies and delegation, e.g., "administrator says that Alice can be trusted when she says to delete $file_1$": Alice speaks for the administrator concerning the deletion of $file_1$, thus she should be trusted as much as the administrator.

In this paper we present a Fibred Security Language (FSL) for access control in distributed systems. Fibring is a general methodology due to Gabbay [1] that aims to combine logics.

Dagstuhl Seminar Proceedings 09121
Normative Multi-Agent Systems
http://drops.dagstuhl.de/opus/volltexte/2009/1908

Suppose we have two different logics $\mathcal{C}$ and $\mathcal{D}$ with languages $\mathbb{L}_{\mathcal{C}}$, $\mathbb{L}_{\mathcal{D}}$ and semantics $S_{\mathcal{C}}$, $S_{\mathcal{D}}$ respectively. Intuitively, the fibring process consists in defining a combined language $\mathbb{L} \supset \mathbb{L}_{\mathcal{C}} \cup \mathbb{L}_{\mathcal{D}}$ together with a new semantics S in which we can evaluate formulas of both $\mathcal{C}$ and $\mathcal{D}$.

From a semantical point of view, logics for distributed access control rely on one of the following approaches

- Operational Semantics [2].
- Declarative Semantics [3,4].
- Classical/Intuitionistic Modal logic [5,6,7,8].

Each view has positive and negative aspects.

Operational Semantics, if rules are wisely crafted, could be extremely clear but very often tractability must be sacrificed for simplicity. SecPAL, for instance, has an extremely clear semantics expressed with just three rules, but in practice they are awkward to employ in evaluating formulas. To overcome this difficulty queries in [2] are evaluated exploiting Datalog that has a stable model semantics which is not clearly related with the rules of the operational semantics.

Logics that rely on declarative semantics have a clearly specified notion of proof of compliance which is strictly based on the framework in which the reasoning is carried out. PROLOG and Datalog seems to be the most used solutions to obtain answer sets from a database of distributed policies. The negative aspect is that using declarative approaches it could be extremely difficult to have a formal "meaning" for every set of policies and credentials, so that one can compute this meaning and inspect whether it is the same as the policy author's intention.

Modal logic have been employed by Abadi [6] to model logics for access control, in this view a logic can be studied through its axiomatization or on the basis of its semantics analyzing how to link models with formulas. One major advantage is that with a clear bound between syntax and semantics the proof of compliance procedure is based on well-understood, formal foundation. A mayor loss is that it could be extremely difficult to compose different logics within a common framework if we do not rely on fibring.

Every approach has some positive aspects that should not be left out in modelling a logic for distributed access control. With FSL we propose a general language to compose (fibring) existing logics on the basis of their semantics, in particular Section 4.2 is devoted to introduce an authorization logic called predicate FSL in which we fibre intuitionistic logic with multimodal logic. Future papers will be devoted to extend and compose existing access control logics (see Section 6).

In predicate FSL we have formulas of the kind

$$\{x\}\varphi(x) \textbf{ says } \psi \tag{1}$$

where $\{x\}\varphi(x)$ represents the group composed by all the principals[4] that satisfy $\varphi(x)$ and ψ is a general formula. We see the **says** as a modality to express that a certain principal supports some statement (see Section 2).

In this view, Formula 1 becomes

$$\Box_{\{x\}\varphi(x)}\psi \tag{2}$$

In which ψ is the statement that the extension of $\varphi(x)$ *as a group* of individuals supports; note also that the modality is indexed by principals. Up to authors knowledge, existing approaches that employ the **says** operator do not offer the possibility to have a first-order formula specifying the principals.

This view on access control logics offers a wide range of expressiveness in defining policies and freedom in crafting logics. In fact we can let $\varphi(x)$ and ψ belong to two different languages $\mathbb{L}_p$ and $\mathbb{L}_e$ as language of principals and security expressions respectively which refers to two different systems (semantics).

For instance we can think of formulas in $\mathbb{L}_p$ be SQL queries and formulas in $\mathbb{L}_e$ be Delegation Logic [3] expressions.

The main problem is to formally specify how to evaluate expressions like 2 and this is the main role of the fibring methodology [1] which, depending on the chosen languages (and systems), must be carefully defined in order to have a combined logic which is coherent and does not collapse.

In this paper, in order to show the full expressiveness of our approach, we decide to make $\mathbb{L}_p = \mathbb{L}_e = \mathbb{L}$, where $\mathbb{L}$ is a classical first order language, whereas the relying system S is intuitionistic modal logic; this is predicate FSL. This approach offers us to iterate the **says** modality and to have extremely complex formulas in which free variables are shared between different levels of nesting of the $\Box$ (see Section 3.1 for examples).

Throughout the paper we will show how with predicate FSL is possible to give answers to the following questions:

1. How to define a general semantic model in order to extend existing security languages?
2. How to make a principal speak for another principal on all formulas without resorting second order languages?
3. How to have groups of principals supporting a sentence expressed by a first-order formula with free variables?
4. How to express chain of delegation by means of the **says** modality and how to constrain delegation depth?
5. How to express separation of duties in a clear and compact way?
6. How to deal with roles in distributed access control?

The paper is structured as follows. First, in Section 2 we discuss which properties of the **says** operator are desirable in logic and which are not, highlighting the dependencies among them in different logics. Second, in Section 3 we consider how to extend the authorization logic on the side of the principals which

[4] Example of principals are: Users, machines, channels, conjunction of principals, groups . . . [6]

can assert **says** statements. Then, we present the basic *fibred security language FSL* in Section 4 and we extend it to predicate logic in Section 4.2. In Section 5 we give a simple example to show how to employ predicate FSL and Section 6 ends the paper.

2 Properties of access control logics

In this section we first summarize how the **says** operator is used in access control logics, and then we discuss which properties are desired for this operator and which are not, showing the dependencies among the different properties in existing logics.

2.1 Access control logics

The access control logic we propose aims at distributed scenarios. Thus, to express delegation among principals, it is centered, like the access control logic of [5,3], on formulas such as "A **says** s" where A represents a principal, s represents a statement (a request, a delegation of authority, or some other utterance), and **says** is a modality. Note that it is possible to derive that A **says s** even when A does not directly utter **s**. For example, when the principal A is a user and one of its programs includes **s** in a message, then we may have A **says s**, if the program has been delegated by A. In this case, A **says s** means that A has caused **s** to be said, that **s** has been said on A's behalf, or that A supports **s**.

We assume that such assertions are used by a reference monitor in charge of making access control decisions for resources, like o. The reference monitor may have the policy that a particular principal A is authorized to perform $Do(o)$. This policy may be represented by the formula: $(A \textbf{ says } Do(o)) \rightarrow Do(o)$, which expresses that A controls $Do(o)$. Similarly, a request for the operation on o from a principal B may be represented by the formula: B **says** $Do(o)$. The goal of the reference monitor is to prove that these two formulas imply $Do(o)$, and grant access if it succeeds. While proving $Do(o)$ the reference monitor does not need that the principal B controls **s**. Rather it may exploit relations between A and B and some other facts. For example, it may knows that B has been delegated by A, and, thus, that B speaks for A as concerns $Do(o)$, in formulas:

$$(B \textbf{ says } Do(o)) \rightarrow (A \textbf{ says } Do(o))$$

This simple example does not show the subtleties arising from the formalization of the **says** operator, since expressing simple properties like controlling a resource or speaking for another principal may imply less desirable properties, leading to security risks, or even to inconsistent or degenerate logic systems [9].

2.2 Modality axioms

The following are some axioms considered in the literature for the operator **says** , in particular by [9]. We discuss whether they are desirable or not, and which are the relationships among them in different logics, in particular, classical and intuitionistic logic. We write A **says** X as $\Box_A X$. A might be an index U and X ranges over formulas.

Definition 1 (Axiom list).

1. B speaks for A *(notation $B \Rightarrow A$):*

$$\forall X[\Box_B X \to \Box_A X].$$

Note that here we are quantifying over formulas but if we take it as an axiom schema for the relation between A and B, this will automatically be universally quantified.
This is the fundamental relation among principals in access control logics. If $B \Rightarrow A$ from the fact that principal B says something means the reference monitor can believe that principal A says the same thing. This relation serves to form chains of responsibility: a program may speak for a user, much like a key may speak for its owner, much like a channel may speak for its remote end-point. In some logics this relation is primitive. The reference monitor's participation is left implicit, as in the all the other axioms.

2. Restricted speaks for

$$\alpha(X) \wedge \Box_B X \to \Box_A X$$

where $\alpha(X)$ be any formula and X a new variable.
Restriction of "speaks for" is similar to the one [10] introduces. In particular, if $\alpha(X) = \varphi \to X$, then the above formula would refer to B speaks for A on all consequences of φ [8].
Other kinds of restrictions can refer to variables occurring in X. We consider such kind of constraints in Section 3.

3. A controls X

$$\Box_A X \to X$$

This axiom is used in other axioms below.

4. Hand-off axiom

$$\Box_A \forall X[\Box_B X \to \Box_A X] \to \forall X[\Box_B X \to \Box_A X]$$

or more briefly:

$$\Box_A (B \Rightarrow A) \to (B \Rightarrow A)$$

Hand-off states that whenever A says that B speaks for A, then B does indeed speak for A. This axiom allows every principal to decide which principals speak on its behalf, since it controls the delegation to other principals.
Sometimes this axiom follows from logic rules as in [9], sometimes it is assumed as an axiom. Note that the general axiom is too powerful, and thus risky for security: for example when A represents a group: if A **controls** *$(B \Rightarrow A)$ then any member of A can add members to A. Thus, for instance, [6] does not adopt this axiom.*

5. Generalised Hand-off
Since A **controls** X *is defined as* $\Box_A X \to X$.
Then

$$\forall XY(A\ \textbf{controls}\ (X \to \Box_A Y))$$

or explicitly

$$\Box_A(X \to \Box_A Y) \to (X \to \Box_A Y)$$

For $X = \Box_B Y$, *we get hand-off:*

$$\Box_A(\Box_B Y \to \Box_A Y) \to (\Box_B Y \to \Box_A Y)$$

Generalised Hand-off is equivalent to Bind (see item 12 below). It follows from logic rules in [9].

6. Dual of Hand-off

$$\Box_A(A \Rightarrow B) \to (A \Rightarrow B)$$

This is implied by Unit in CDD [9], where it is equivalent to Unit axiom if there is a truth telling principal.

7. Least privilege

$$(X \to Y) \to (\Box_A X \to \Box_A Y)$$

"Every program and every user of the system should operate using the least set of privileges necessary to complete the job" [11].

8. Ordinary modal axioms
- Closure under consequence

$$\Box_A X \wedge \Box_A(X \to Y) \to \Box_A Y$$

- Necessitation

$$\vdash X \textit{ implies } \vdash \Box_A X$$

9. Axiom C4

$$\Box_A \Box_A X \to \Box_A X$$

10. Escalation

$$\Box_A X \to X \vee \Box_A \bot$$

Escalation is not considered as a desirable property. Thus we must be careful that it does not follow from other properties (like from Unit or Bind in classical logics). It amounts to "if A **says** s *then* s *or*A **says** $false$*": from* A **says** s *may follow a statement "much falser" than* s*. As an example of its riskiness, consider that from* (A **controls** s) $\wedge$ (B **controls** s) *it allows to infer that if* A **says** B **says** s *then* s *follows. If the logic is not able to avoid escalation, the only cumbersome solution is to make* A *avoid saying that* B *says* s *unless he really wishes to say* s*.*
Unit and Bind together do not imply Escalation in CDD [9], while Escalation implies Bind. In classical logic, Unit implies Escalation while Escalation does not imply Bind.

11. Unit

$$X \to \Box_A X$$

Unit is stronger than the necessitation rule. In classical logic, adopting Unit implies that each principal either always says the truth or it says false: $(A \to B) \vee (B \to A)$*. In the first case* A *speaks for any other principal, in the latter any other speaks for* A*. The policies described by this kind of systems are too manicheist.*

12. Bind

$$(X \to \Box_A Y) \wedge \Box_A X \to \Box_A Y$$

Abadi [9] provides an example of discussion about the implications of the different axioms of access control logics.

According to [9] in classical logic, Bind is equivalent to escalation and Unit implies Escalation. Intermediate systems requiring C4 do not lead to escalation, but they are not sufficient for modelling delegation.

To solve this problem Abadi in [9] introduces CDD, a second-order propositional intuitionistic logic; in Section 4.2 we present predicate FSL which extends CDD expressiveness without using a second-order language.

3 Reasoning about principals

In the previous section we considered the properties of the **says** operator keeping the principal indexing the modality as a propositional atom[5]. In this section we make a further step towards predicate FSL taking into account how to express the key properties of access control policies in the proposed language.

3.1 FSL: An extended logic of principals

The logic we propose uses a construct which allows to build principals using general logic formulas: $\{x\}\varphi(x)$ **says** ψ. In this section we will show how we can exploit it. Note that $\varphi(x)$ and ψ can share variables and φ may include occurrences of the **says** operator. Notice that x can occur in ϕ but then this occurrence is not related to the x in $\{x\}\varphi(x)$. The formula $\{x\}\varphi(x)$ is used to select the set of principals making the assertion **says**.

To select a single principal whose name is A we do:

$$\{x\}(x = A) \textbf{ says s}$$

We write A **says s** for $\{x\}(x = A)$ **says s**, where A is an individual principal.

The following formula means that all $user$s together ask to delete $file_1$:

$$\{x\}user(x) \textbf{ says } delete(file_1)$$

[5] Up to authors knowledge, like all existing formal access control logics do.

Since $\varphi(x)$ and ψ can share variables, we can put restrictions on the variables occurring in ψ. E.g., the set of all users who all own file(s) y asks to delete the file(s) y.

$$\{x\}(user(x) \wedge own(x, y)) \textbf{ says } delete(y)$$

However, the formula above is satisfactory only in the particular situation where we are talking about the set of all users who assert **says** at once as a group (committee).

We can as well express that each member of a set identified by a formula can assert **says** separately. E.g., each user deletes individually the files he owns:

$$\forall x(user(x) \wedge own(x, y)) \rightarrow \{z\}(z = x) \textbf{ says } delete(y)$$

Note that the latter formula usually implies the former but not vice versa[6]. The former formula,

$$\{x\}(\text{user}(x) \wedge \ \text{own}(x, y)) \textbf{ says } \ \text{delete}(y)$$

expresses the fact that the group of users who own y, i.e. all the owners of y decide (or **say**) as a group to delete y. So maybe they called a meeting, discussed the matter and then had a secret vote. The majority voted to delete y but some voted not to delete. The group outcome was to delete.

The second formula

$$\forall x(\text{user}(x) \wedge \ \text{own}(x, y) \rightarrow \{z\}(z = x) \textbf{ says } \ \text{delete}(y))$$

expresses the fact that each user who owns y **says** to delete it. This usually implies that the users as a group would **say** to delete y but not necessarily (see footnote 3). Concerning the majority vote example, it may be the case that it is impossible to convene enough users to have a vote and so the set of users never manages to " **say** " as a group to delete y.

Operations on principals We can express the fact that two principals A and B together says **s** :

$$\{x\}(x = A \vee x = B) \textbf{ says s}$$

which corresponds to

$$\{A, B\} \textbf{ says s}$$

[6] In fact, it could be sensible to have situations in which if all the members of a group say something then the whole group says it but not vice versa.

$$\forall t(\varphi(t) \rightarrow t \textbf{ says } \psi) \rightarrow \{x\}\varphi(x) \textbf{ says } \psi$$

For instance, a committee may approve a paper that not all of its members would have accepted.

If we want to express that the intersection of two different kind of principals (T_1,T_2) **says** ψ:

$$\exists x(T_1(x) \wedge T_2(x)) \rightarrow \{y\}(y = x) \textbf{ says } \psi^{7}$$

with this approach we can also have negation in selecting principals:

$$\{x\}(x \neq A) \textbf{ says s}$$

Variables over principals The possibility to express principals as variables allows first of all attribute-based (as opposed to identity-based) authorization as in [2]. Attribute-based authorization enables collaboration between parties whose identities are initially unknown to each other. The authority to assert that a subject holds an attribute (such as being a student) may then be delegated to other parties, who in turn may be characterised by attributes rather than identity. In the example below, a shop gives a discount to students. The authority over the student attribute is delegated to holders of the university attribute, and authority over the university attribute is delegated to known principal, the Board of Education.

Shop says x is entitled to discount if x is a student.

$$\begin{gathered} \mathit{Shop} \textbf{ says } (\mathit{student}(x) \rightarrow \\ \{y\}(x = y) \textbf{ controls } \mathit{discount}) \end{gathered}$$

Shop says x can say z is a student if x is a university

$$\begin{gathered} \mathit{Shop} \textbf{ says } (\mathit{university}(x) \rightarrow \\ \{y\}(x = y) \textbf{ controls } \mathit{student}(z)) \end{gathered}$$

Shop says BoardOfEducation can say x is a university

$$\mathit{Shop} \textbf{ says } (BoardOfEducation \textbf{ controls } \mathit{university}(x))$$

We may have more complicated policies involving more that two principals, like in the following example [3].

$$\begin{gathered} \{y\}(y = A) \textbf{ says } (((\{y\}(y = C) \textbf{ says } fraudulent(x)) \wedge \\ \{y\}(y = D) \textbf{ says } \mathit{expert}(C)) \rightarrow fraudulent(x)) \end{gathered}$$

Since φ in $\{x\}\varphi(x)$ **says** ψ can be any formula, it can contain even occurrences of the **says** operator. This allows to refer to principals who made previous assertions of the **says** operator. For example, we can express the following: the members of the board who said to write a file they own, ask to delete it.

In symbols

$$\begin{gathered} \{x\}[\{u\}\mathit{member\text{-}board}(u) \textbf{ says } ((\mathit{member\text{-}board}(x) \wedge \\ f\mathit{ile\text{-}owner}(y, x)) \rightarrow \mathit{write}(y))] \\ \textbf{says } \mathit{delete}(y) \end{gathered}$$

[7] For instance, T_1 could be *club_member* and T_2 *adult*.

Like in [2] delegation can be restricted to principals respecting some requirements: Fileserver is a trusted principal who delegates file reading authorizations only to the owners of files:

$$\forall x\ own(x,y) \rightarrow (Fileserver\ \textbf{says}\ (\{z\}(z=x) \\ \textbf{says}\ read(y) \rightarrow Fileserver\ \textbf{says}\ read(y)))$$

Variables over principals allow width-bounded delegation. Suppose A wants to delegate authority over *is a friend* fact to Bob. She does not care about the length of the delegation chain, but she requires every delegator in the chain to satisfy some property, e.g. to possess an email address. Principals with the *is a delegator* attribute are authorized by A to assert *is a friend* facts, and to transitively re-delegate this attribute, but only amongst principals with a matching email address.

A says x can say y is a friend if x is a delegator

$$A\ \textbf{says}\ ((delegator(x) \rightarrow \\ (\{y\}(x=y)\ \textbf{says}\ friend(z))) \rightarrow friend(z)$$

A says B is a delegator

$$A\ \textbf{says}\ delegator(B)$$

A says x can say y is a delegator if x is a delegator, y possesses email.

$$A\ \textbf{says}\ ((delegator(x) \wedge has\text{-}email(y)) \rightarrow \\ (\{w\}(w=x)\ \textbf{controls}\ delegator(y)))$$

As with depth-bounded delegation, this property cannot be enforced in SPKI/SDSI, DL or XrML.

Restrictions on says Another issue concerns restrictions on speaks for on some issues. Some authors restrict $\Rightarrow$ to a set of propositions [15]

$P \Rightarrow_T Q$ means that the proposition **s** in P **says s** $\rightarrow Q$ **says s** must belong to T.

We can put some restrictions on the variables:

$$(\{x\}(user(x) \wedge owns(x,y))\ \textbf{says}\ delete(y)) \rightarrow \\ (\{z\}(super\text{-}user(z))\ \textbf{says}\ delete(y)$$

Moreover we can use the following to restrict the scope of speaks for:

$$\alpha(X) \wedge \Box_B X \rightarrow \Box_A X$$

If $\alpha(X) = \varphi \rightarrow X$ then B speaks for A only on consequences of φ.

The restricted speaks for is strictly related with delegation, if for instance $B \Rightarrow_T A$ we say that B is delegated by A on T. If we want to limit the delegation chain to one step such that we do not permit B to delegate another principal C on T, we add the following constraint:

$$(C \Rightarrow_T B \Rightarrow_T A) \rightarrow (C = B)$$

Separation of duties One of the main concerns in security is the separation of duties: for example the principal(s) signing an order cannot be the same principals who approve it:

$$\neg(\{x\}(\{y\}(x = y) \textbf{ says } \mathit{sign}(\mathit{project})) \textbf{ says} \\ \mathit{approving}(\mathit{project}))$$

In this formula we exploit the full potentiality of FSL in that the principal is defined in terms of the **says** operator.

As noticed in [2] separation of duties requires using negation.

Roles When roles are considered, it emerges the question whether we consider roles types or instances. We distinguish here among roles instances which can be principals by themselves or properties of other principals. So a sentence like "*A*, who plays a role *x* of type *R*, **says s**" becomes:

$$\forall x(x = A \wedge \mathit{role\text{-}played\text{-}by}(x, y) \wedge R(y)) \rightarrow \\ \{z\}(z = y) \textbf{ says s}$$

As concerns hierarchies:

$$\forall x \ \mathit{super\text{-}user}(x) \rightarrow \mathit{user}(x)$$

then

$$\forall x \ \mathit{super\text{-}user}(x) \rightarrow (\{z\}(x = z) \textbf{ says s}) \rightarrow \\ (\forall x \ \mathit{user}(x) \rightarrow (\{z\}(x = z) \textbf{ says s})$$

Instead

$$(\{x\}\mathit{super\text{-}user}(x) \textbf{ says s}) \rightarrow (\{x\}\mathit{user}(x) \textbf{ says s})$$

is less useful: if all super-users say **s** than all users say **s**.

In Abadi [6] if *A* says something in a role, then it is true that he is playing a role. However, he admits that there should be some requirements to play a role.

For instance, we require that a super-user is a technician:

$$\forall x \ \mathit{super\text{-}user}(x) \rightarrow \mathit{technician}(x)$$

then we can say

$$\forall x \ (x = A \wedge \mathit{super\text{-}user}(x)) \rightarrow (\{z\}(x = z) \textbf{ says s})$$

but there can be no super-user *x* if *A* is not a technician.

Parameterized roles can add significant expressiveness to a role-based system and reduce the number of roles [2,13,14]. If we model roles as instances they can have attributes. For instance the example in [2] "NHS[8] says x can access health record of patient if x is a treating clinician of patient" can be modeled as:

[8] National Health Service.

$$\begin{gathered}(\mathit{clinician\text{-}role}(x) \wedge \mathit{patient}(p) \wedge \mathit{record}(r,p) \wedge \\ \mathit{treats}(x,p)) \rightarrow \\ (\{w\}(w = x) \textbf{ says } \mathit{access}(r) \rightarrow \mathit{NHS} \textbf{ says } \mathit{access}(r)))\end{gathered}$$

The operator used to represent a principal A in the role B ($A \mid B$) in [6] is modeled in this way.

$$(A \mid B) \textbf{ says s } \equiv A \textbf{ says } (B \textbf{ says s})$$

In order to match the predicate role-played-by with the above definition we can add the following (where x is a role):

$$\begin{gathered}\forall x, y \ \mathit{role\text{-}played\text{-}by}(x,y) \rightarrow \\ ((x \textbf{ says s}) \rightarrow \\ y \textbf{ says } (\{z\}(z = x) \textbf{ says s}))\end{gathered}$$

Discretionary access control Discretionary access control allows users to pass on their access rights to other users at their own discretion. For instance we can express: "FileServer says user can say x can access resource if user can access resource"[2]

$$\begin{gathered}\forall x \ \mathit{user}(x) \wedge \mathit{user}(z) \rightarrow (\mathit{Fileserver} \textbf{ says} \\ \{w\}(\{y\}(w = y = x) \textbf{ controls } \mathit{access}(u)) \textbf{ controls} \\ \{t\}(t = z) \textbf{ controls } \mathit{access}(u))\end{gathered}$$

Groups In FSL you have to possibility to express how the set $\{x|\varphi(x) \text{ holds}\}$ says what it says, e.g. If $\varphi(x) = (x = A_1) \vee (x = A_2) \vee (x = A_3)$ then if at least one of $\{A_i\}$ **says** ψ is enough for the group to **say** ψ we add:

$$\{x\}\varphi(x) \textbf{ says } \psi \leftrightarrow \bigvee_i \{x\}(x = A_i) \textbf{ says } \psi.$$

This represents the fact that each principal in the group can speak for the whole group. We can as well express that every group has a spokesman (maybe several ones dependent on issues), that one cannot be a spokesman for two different groups and that a group controlling an issue cannot control issues inconsistent with the definition of the group. We can define groups using what they say as part of the definition, put restriction on what they further say or control.

1. *Every group has a spokesman.*
   This is an axiom schema in φ. Let **spoke**(φ, y) be

   $$\begin{gathered}\textbf{spoke}(\varphi, y) = (\forall X[\{x\}\varphi(x) \textbf{ says } X \leftrightarrow \\ \{x\}(x = y) \textbf{ says } X])\end{gathered}$$

   We then take the axiom as $\exists y$ **spoke** (φ, y).

2. *One cannot be a spokesman for two different groups.*

$$\forall y[\ \textbf{spoke}\ (\varphi_1, y) \wedge\ \textbf{spoke}\ (\varphi_2, y) \to \\ \forall x[\varphi_1(x) \leftrightarrow \varphi_2(x)]]$$

3. A group cannot control issues inconsistent with the definition of the group

$$\frac{\vdash \varphi \wedge \psi \to \bot}{\vdash [(\{x\}\varphi(x)\ \textbf{says}\ \psi) \to \psi] \to \bot}$$

The following additional axiom expresses that the group identified by the extension of $\{x\}\varphi(x)$ says ψ if at least two members says ψ:

$$\{x\}(\bigvee_i x = A_i)\ \textbf{says}\ \psi \text{ iff} \\ \bigvee_{i \neq j}[\{x\}(x = A_i)\ \textbf{says}\ \psi \wedge \{x\}(x = A_j)\ \textbf{says}\ A_j]$$

More generally, majority voting in $\{x\}\varphi(x)$ **says** ψ, is just an axiom.

$$\{x\}\varphi(x)\ \textbf{says}\ \psi \leftrightarrow \bigvee_i \{x\}\varphi_i(x)\ \textbf{says}\ \psi$$

where $\varphi_i(x)$ are all formulas $(\forall x \varphi_i(x) \to \varphi(x))$ defining majorities in the set $\{x\}\varphi(x)$.

Majority vote is an example of threshold-constrained trust SPKI/SDSI [12]. The concept of k-of-n threshold subjects means that at least k out of n given principals must sign a request and it is used to provide a fault tolerance mechanism. RTT has the language construct of "threshold structures" for similar purposes [39]. As in SecPAL [2] there is no need for a dedicated threshold construct, because threshold constraints can be expressed directly.

4 The basic system FSL

This section introduces our basic system FSL step by step from a semantics point of view. First we introduce modalities indexed by propositional atoms, then we take into account classical and intuitionistic models for the propositional setting and finally we give semantics to predicate FSL that we extensively employed in previous sections.

This system can be defined with any logic $\mathbb{L}$ as a *Fibred Security System based on* $\mathbb{L}$. We will motivate the language for the cases of $\mathbb{L}$ = classical logic and $\mathbb{L}$ = intuitionistic logic.

Basically adding the **says** connective to a system is like adding many modalities. So to explain and motivate FSL technically we need to begin with examining options for adding modalities to $\mathbb{L}$. Subsection 4.1 examines our options of how to add modalities to classical and intuitionistic logics. The presentation and discussion is geared towards subsection 4.2 which presents FSL.

4.1 Adding modalities

We start by adding modalities to classical propositional logic. We are going to do it in a special way. The reader is invited to closely watch us step-by-step,

Our approach is semantic.

Let S be a nonempty set of possible worlds. For every subset $U \subseteq S$ consider a binary relation $R_U \subseteq S \times S$.

This defines a multi-modal logic, containing K modalities $\Box_U, U \subseteq S$. The models are of the form $(S, R_U, t_0, h), U \subseteq S$. In this view, if $U = \{t | t \vDash \varphi_U\}$ for some φ_U we get a modal logic with modalities indexed by formulas of itself. This requires now a formal definition.

Definition 2 (Language). *Consider (classical or intuitionistic) propositional logic with the connectives $\wedge, \vee, \to, \neg$ and a binary connective $\Box_\varphi \psi$, where φ and ψ are formulas.*[9] *The usual definition of wff is adopted.*

Definition 3. *We define classical Kripke models for this language.*

1. *A model has the form*

$$\mathbf{m} = (S, R_U, t_0, h), U \subseteq S$$

where for each $U \subseteq S, R_U$ is a binary relation on S. $t_0 \in S$ is the actual world and h is an assignment, giving for each atomic q a subset $h(q) \subseteq S$.

2. *We can extend h to all formulas by structural induction:*
   - *$h(q)$ is already defined, for q atomic*
   - $h(A \wedge B) = h(A) \cap h(B)$
   - $h(\neg A) = S - h(A)$
   - $h(A \to B) = (S - h(A)) \cup h(B)$
   - $h(A \vee B) = h(A) \cup h(B)$
   - $h(\Box_\varphi \psi) = \{t |$ *for all* s $(tR_{h(\varphi)}s \to s \in h(\psi))\}$
3. $\mathbf{m} \vDash A$ *iff* $t_0 \in h(A)$.

There is nothing particularly new about this except possibly the way we are looking at it.

Let us now do the same for intuitionistic logic. Here it becomes more interesting. An intuitionistic Kripke model has the form

$$\mathbf{m} = (S, \leq, t_0, h),$$

where $(S, \leq)$ is a partially ordered set, $t_0 \in S$ and h is an assignment to the atoms such that $h(q) \subseteq S$. We require that $h(q)$ is a closed set, namely

- $x \in h(q)$ and $x \leq y$ imply $y \in h(q)$.

[9] There are many such connectives, e.g. φ **says** $\psi, \varphi > \psi$ (conditional), $\bigcirc(\varphi/\psi)$ relative obligation, etc. The semantics given to it will determine its nature.

Let D be a set and we can add for each $U \subseteq D$ a binary relation R_U on S. This semantically defines an intuitionistic modality, $\Box_U$.

In intuitionistic models we require the following condition to hold for each formula A, i.e. we want $h(A)$ to be closed:

- $x \in h(A)$ and $x \leq y \Rightarrow y \in h(A)$

This condition holds for A atomic and propagates over the intuitionistic connectives $\wedge, \vee, \rightarrow, \neg, \bot$. To ensure that it propagates over $\Box_U$ as well, we need an additional condition on R_U. To see what this condition is supposed to be, assume $t \vDash \Box_U A$. This means that

$$\forall y(tR_U y \Rightarrow y \vDash A).$$

Let $t \leq s$. If $s \nvDash \Box_U A$, then for some z such that $sR_U z$ we have $z \nvDash A$. This situation will be impossible if we require

$$t \leq s \wedge sR_U z \Rightarrow tR_U z. \qquad (*)$$

Put differently, if we use the notation:

$$R'_U(x) = \{y|xR_U y\}$$

then

$$x \leq x' \Rightarrow R'_U(x) \supset R'_U(x'). \qquad (*)$$

So we now talk about modalities R_U, for $U \subseteq S$. We ask what happens if U is defined by a formula φ_U, i.e. $U = h(\varphi_U)$. This will work only if U is closed

- $t \in U \wedge t \leq s \Rightarrow s \in U$.

So from now on, we talk about modalities associated with closed subsets of S.

We can now define our language. This is the same as defined in Definition 2. We now define the semantics.

Definition 4. *A model has the form*

$$\mathbf{m} = (S, \leq, R_U, t_0, h), U \subseteq S$$

where $(S, \leq)$ is a partial order, $t_0 \in S$, and each $U \subseteq S$ is a closed set and so is $h(q)$ for atomic q. R_U satisfies condition (*) above. We define the notion $t \vDash A$ for a wff by induction, and then define

$$h(A) = \{t|t \vDash A\}.$$

So let's define $\vDash$:

- $t \vDash q$ *iff* $t \in h(q)$
- $t \vDash A \wedge B$ *iff* $t \vDash A$ *and* $t \vDash B$
- $t \vDash A \vee B$ *iff* $t \vDash A$ *or* $t \vDash B$
- $t \vDash A \rightarrow B$ *iff for all* $s, t \leq s$ *and* $s \vDash A$ *imply* $s \vDash B$

- $t \vDash \neg A$ *iff for all* s*,* $t \leq s$ *implies* $s \nvDash A$
- $t \nvDash \bot$
- $t \vDash \Box_\varphi \psi$ *iff for all* s *such that* $tR_{h(\varphi)}s$ *we have* $s \vDash \psi$*. We assume by induction that* $h(\varphi)$ *is known.*
- $\mathbf{m} \vDash A$ *iff* $t_0 \vDash A$*.*

It is our intention to read $\Box_\varphi \psi$ as φ **says** ψ.

Example 1 (Two intuitionistic modalities). Let us examine the case of two intuitionistic modalities in more detail Let us call them $\Box_A$ and $\Box_B$ and their accessibility relations R_A and R_B. So our Kripke model has the form $(S, \leq, R_A, R_B, t_0, h)$. We know for $\mu = A$ or $\mu = B$ that we have in the model

$$t \leq s \wedge sR_\mu z \to tR_\mu z. \tag{$*$}$$

What other conditions can we impose on $\Box_\mu$?

1. *The axiom* $X \to \Box_\mu X$
   This corresponds to the condition

   $$xR_\mu y \to x \leq y \tag{$*1$}$$

2. *The axiom* $\Box_B X \to \Box_A X$
   This corresponds to the condition

   $$xR_A y \to xR_B y \tag{$*2$}$$

3. Note that $\Box_B X \to \Box_A X$ is taken in (*2) as an axiom schema. If we want to have $t \vDash \forall X(\Box_B X \to \Box_A X)$ i.e. we want $\Box_B \varphi \to \Box_A \varphi$ to hold at the point $t \in S$ for all wff φ, we need to require (*2) to hold above t, i.e.

   $$\forall x, y(t \leq x \wedge xR_A y \to xR_B y) \tag{$*3$}_t$$

4. Consider now an axiom called *hand-off A to B*.

   $$\Box_A(\forall X(\Box_B X \to \Box_A X)) \to \forall X(\Box_B X \to \Box_A X)$$

   This axiom has a second order propositional quantifier in it.
   The antecedent of the axiom wants $\Box_A(\forall X \Box_B X \to \Box_A X))$ to hold at t_0. This means in view of (3) above that $(*4_a)$ needs to hold

   $$\forall t(t_0 R_A t \to (*3)_t) \tag{$*4_a$}$$

   The axiom says that if the antecedent holds at t_0 so does the consequent, i.e.

   $$t_0 \vDash \forall X(\Box_B X \to \Box_A X).$$

   We know the condition for that to hold is $(*3)_{t_0}$. Thus the condition for Hand-off A to B is

   $$\forall t[t_0 R_A t \to (*3)_t] \to (*3)_{t_0} \tag{$*4$}$$

The important point to note is that although the axiom is second order (has $\forall X$ in it both in the antecedent and consequence), the condition on the model is first order[10].

5. There is another modal axiom called escalation for A

$$\Box_A X \to X \vee \Box_A \bot$$

The condition for that is

$$\exists y(xR_Ay) \to xR_Ax \tag{$*5$}$$

To check whether we can have hand-off from A to B without escalation for A, for some choice of R_A and R_B, we need to check whether we can have (*4) without having (*5), for some wise choice of R_A and R_B.

6. Consider a Kripke model $(S, \leq, t_0)$ which is nonending and dense, i.e.
   – $\forall x \exists y(x \lneqq y)$
   – $\forall xy(x \lneqq y \to \exists z(x \lneqq z \lneqq y))$
   In this model let

$$xR_Ay \text{ be } x \lneqq y$$
$$xR_By \text{ be } x \leq y.$$

We have here that $(*3)_t$ holds for any t because it says

$$\forall xy(t \leq x \wedge x \lneqq y \to x \leq y)$$

Therefore (*4) also holds. This is hand-off from A to B.
However, escalation does not hold because

$$\exists y(x \lneqq y) \to x \lneqq x$$

is false.

7. *α relative hand-off*
   Let $\alpha(x)$ be any formula and X a new variable. We can write a relative speak axiom.

$$\alpha(X) \wedge \Box_B X \to \Box_A X \tag{$*7$}$$

In particular, if $\alpha(x) = \varphi \to x$, then (*7) would refer to B speaks for A on all consequences of φ.

Definition 5. *Let $(S, \leq, t_0, h)$ be a Kripke model. By a modal function* $\mathbf{E}$ *we mean a function giving to each point $t \in S$ a set of points $\mathbf{E}(t)$ such that*

1. *$t \lneqq s$ for all $s \in \mathbf{E}(t)$.*
2. *$t_1 \leq t_2 \to \mathbf{E}(t_1) \leq \mathbf{E}(t_2)$ where $\mathbf{E}(t_1) \leq \mathbf{E}(t_2)$ means $\forall x \in \mathbf{E}(t_2) \exists y \in \mathbf{E}(t_1)(y \leq x)$.*

Definition 6. $(S, \leq, t_0, \mathbf{E})$ *is* $\mathbf{E}$*-dense iff the following holds:*

[10] Notice that we use first-order but we get a language more expressive than CDD[9] which is second-order.

– *If* $x \in \mathbf{E}(t)$*, then for some* $y, y \in \mathbf{E}(t) \wedge x \in \mathbf{E}(y)$*.*

To show the existence of dense systems, we do the following construction:

1. *Start with* $(S_0, \leq_0, \mathbf{E}_0)$ *where* $\mathbf{E}_0(x)$ *is dense. For example,* $(S_0, \leq_0)$ *may be linear and* $\mathbf{E}_0$ *is generated by a strictly increasing function* $\mathbf{f}$*, i.e.* $\mathbf{E}(x) = \{y|\mathbf{f}(x) \leq y\}$*.*
2. *For every pair of points* $x \lneqq_0 y$*, add the point* (x, y)*.*
Let $x \lneqq_0 (x, y) \lneqq_0 y$ *and let* $S_1 = S_0 \cup \{(x, y)|x \lneqq_0 y\}$*, let* $\leq_1$ *be transitive closure of* $\leq_0 \cup \{(x, (x, y)), ((x, y), y)\}$*. Let* $\mathbf{E}_1$ *be defined from* $\leq_1$ *as follows: First let* $\bar{\mathbf{E}}$ *denote the* $\leq_1$ *closure of* $\mathbf{E}$*.*
$$x \in \bar{\mathbf{E}} \text{ iff } \exists y \in \mathbf{E} \text{ such that } y \leq_1 x.$$
Second, let for each x
$$P(x) = \{(x, y)|x \leq_0 y \text{ and } x \neq y\}$$
Then let
$$\mathbf{E}_1(x) = \overline{\mathbf{E}_0(x) \cup P(x)}$$
$$\mathbf{E}_1((x, y)) = \{z|y \leq_1 z\}.$$
3. *Let* $(S_{n+1}, \leq_{n+1}, t_0, \mathbf{E}_{n+1})$ *be obtained from* $S_{n+1}, \leq_{n+1}$ *in the same way as in step 2.*
4. *Let* $(S_\infty, \leq_\infty, t_0, \mathbf{E}_\infty)$ *be the union.*
$$S_\infty = \bigcup_n S_n, \leq_\infty = \bigcup_n \leq_n, \mathbf{E}_\infty = \bigcup_n \mathbf{E}_n.$$
Then we have density.
If $y \in \mathbf{E}_\infty(x)$ *then for some* z
$$z \in \mathbf{E}_\infty(x) \wedge y \in \mathbf{E}_\infty(z)$$

Remark 1. In models of Definition 6 we have *Unit* and *C4* hold but not necessarily *Escalation* nor *Generalised Hand-off*.

4.2 Predicate FSL

Intuitively, a predicate FSL fibred model is represented by a set of models linked togheter by means of a *fibring function*, every model has an associated domain D of elements together with a set of formulas that are true in it. In the FSL meta-model, the evaluation of the generic formula $\{x\}\varphi(x)$ **says** is carried out in two steps, first evaluating φ and then ψ in two different models. Suppose $\mathbf{m_1}$ is our (first order) starting model in which we identify $U \subseteq D$ as the set of all the elements that satisfy φ. Once we have U we can access one or more worlds depending on the *fibring function* $\mathbf{f} : \mathcal{P}(D) \rightarrow \mathcal{P}(M)$ which goes from sets of elements in domain D to sets of models. A this point, for every model $\mathbf{m_i} \in \mathbf{f}(U)$ we must check that ψ is *true*, if this is the case then α is true in the meta-model.

The fact that in the same expression we evaluate different sub-formulas in different models it is not completely counter intuitive, for instance, think about a group of administrators that have to set up security policies for their company. From a semantical point of view, if we want to check if ψ holds in the depicted configuration by the administrators, we must

1. Identify all the admins (all the elements that satisfy $admin(x)$).
2. Access the model that all the admins as a group have depicted.
3. Check in that model if ψ is *true* or *false*

Let $\mathbb{L}$ denote classical or intuitionistic predicate logic.[11] We assume the usual notions of variables, predicates, connectives $\wedge, \vee, \rightarrow, \neg$, quantifiers $\forall, \exists$ and the notions of free and bound variables.

Let $\mathbb{L}^+$ be $\mathbb{L}$ together with two special symbols:

- A binary (modality), x **says** y
- A set-binding operator $\{x\}\varphi(x)$ meaning the set of all x such that $\varphi(x)$

Note that semantically at the appropriate context $\{x\}\varphi(x)$ can behave like $\forall x \varphi(x)$ and sometimes in other contexts, we will use it as a set.

Definition 7. *The language FSL has the following expressions:*

1. *All formulas of $\mathbb{L}^+$ are level 0 formulas of FSL.*
2. *If $\varphi(x)$ and ψ are formulas of $\mathbb{L}^+$ then $\alpha = \{x\}\varphi(x)$ **says** ψ are level 1 'atomic' formulas of FSL. If $(x, x_1, \ldots, x_n)$ are free in φ and $y_1, \ldots, y_m$ are free in ψ then $\{x_1, \ldots, x_n, y_1, \ldots, y_m\}$ are free in α. The variable x in φ gets bound by $\{x\}$. The formula of level 1 are obtained by closure under the connectives and quantifiers of $\mathbb{L}^+$.*
3. *Let $\varphi(x)$ and ψ be formulas of FSL of levels r_1 and r_2 resp., then $\alpha = \{x\}\varphi$ **says** ψ is an 'atomic' formula of FSL of level $r = \max(r_1, r_2) + 1$.*
4. *Formulas of level n are closed under classical logic connectives and quantifiers of all 'atoms' of level $m \leq n$.*

Definition 8 (FSL classical fibred model of level n).

1. *Any classical model with domain D is an FSL model of level 0.*
2. *Let $\mathbf{m}$ be a classical model of level 0 with domain D and let for each subset $U \subseteq D, \mathbf{f}^n(U)$ be a family of models of level n (with domain D). Then $(\mathbf{m}, \mathbf{f}^n)$ is a model of level $n + 1$.*

Definition 9 (Classical satisfaction for FSL). *We define satisfaction of formulas of level n in classical models of level $n' \geq n$ as follows.*

*First observe that any formula of level n is built up from atomic predicates of level 0 as well as 'atomic' formulas of the form $\alpha = \{x\}\varphi(x)$ **says** ψ, where φ and ψ are of lower level.*

[11] Classical predicate logic and intuitionistic predicate logic have the same language. The difference is in the proof theory and in the semantics.

We therefore first have to say how we evaluate $(\mathbf{m}, \mathbf{f}^n) \vDash \alpha$.

We assume by induction that we know how to check satisfaction in $\mathbf{m}$ *of any* $\varphi(x)$, *which is of level* $\leq n$.

We can therefore identify the set $U = \{d \in D \mid \mathbf{m} \vDash \varphi(d)\}$.

Let $\mathbf{m}' \in \mathbf{f}^n(U)$. *We can now evaluate* $\mathbf{m}' \vDash \psi$, *since* ψ *is of level* $\leq n - 1$.

So we say

$$(\mathbf{m}, \mathbf{f}^n) \vDash \alpha \textit{ iff for all } \mathbf{m}' \in \mathbf{f}^n(U), \textit{ we have } \mathbf{m}' \vDash \psi.$$

We need to add that if we encounter the need to evaluate $\mathbf{m} \vDash \{x\}\beta(x)$, *then we regard* $\{x\}\beta(x)$ *as* $\forall x \beta(x)$.

Example 2. Figure 1 is a model for

$$\alpha(y) = \{x\}[\{u\}B(u) \textbf{ says } (B(x) \to A(x,y))] \textbf{ says } F(y)$$

In Figure 1, $\mathbf{m}_1$ is a single model in $\mathbf{f}^1(U_B)$ and $\mathbf{m}_3$ is a single model in $\mathbf{f}^1(U_{E(y)})$, as defined later.

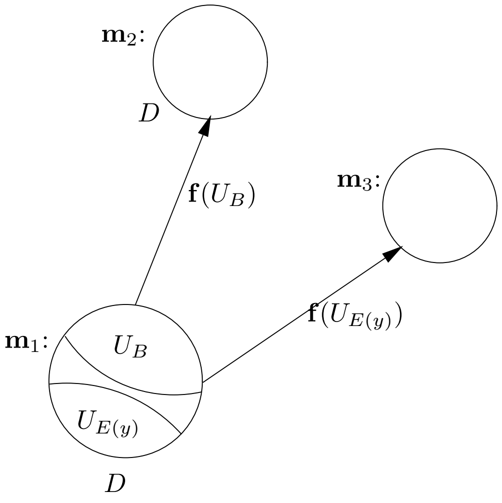


Fig. 1.

The set U_B is the extension of $\{x\}B(x)$ in $\mathbf{m}_1$.

To calculate the set of pairs (x, y) such that $E(x, y) = \{u\}B(u)$ **says** $(B(x) \to A(x, y))$ holds in $\mathbf{m}_1$, we need to go to $\mathbf{m}_2$ in $\mathbf{f}(U_B)$ and check whether $B(x) \to A(x, y)$ holds in $\mathbf{m}_2$, x, y are free variables so we check the value under fixed assignment.)

We now look at $E(y) = \{x\}E(x, y)$ for y fixed, we collect all elements d in D such that $\mathbf{m}_2 \vDash B(d) \to A(d, y)$. Call this set $U_{E(y)}$.

To check $\alpha(y) = \{x\}E(x, y)$ **says** $F(y)$ in $\mathbf{m}_1$ we have to check whether $F(y)$ holds in $\mathbf{m}_3$.

We now define intuitionistic models for FSL. This will give semantics for the intuitionistic language.

Definition 10. *We start with intuitionistic Kripke models which we assume for simplicity have a constant domain. The model* $\mathbf{m}$ *has the form* $(S, \leq, t_0, h, D)$ *where* D *is the domain and* $(S, \leq, t_0)$ *is a partial order with first point* t_0 *and* h *is an assignment function giving for each* $t \in S$ *and each* m*-place atomic predicate* P *a subset* $h(t, P) \subseteq D^m$ *such that* $t_1 \leq t_2 \Rightarrow h(t_1, P) \subseteq h(t_2, P)$

We let $h(P)$ *denote the function* $\lambda t\ h(t, P)$*. For* $t \in S$ *let*

$$\begin{array}{l} S_t = \{s \mid t \leq s\} \\ h(t, P) = h(P) \restriction S_t \\ \leq_t = \leq \restriction S_t \end{array}$$

Let $\mathbf{m}_t = (S_t, \leq_t, t, h_t, D)$*.*

Note that a formula φ *holds at* $\mathbf{m} = (S, \leq, t_0, h, D)$ *iff* $t_0 \vDash \varphi$ *according to the usual Kripke model definition of satisfaction.*

1. *A model of level 0 is any model* $\mathbf{m}$*:* $\mathbf{m} = (S, \leq, t_0, h, D)$*.*
2. *Suppose we have defined the notion of models of level* $m \leq n$*, (based on the domain* D*).*

We now define the notion of a model of level $n + 1$

Let $\mathbf{m}$ *be a model of level 0 with domain* D*. We need to consider not only* $\mathbf{m}$ *but also all the models* $\mathbf{m}_t = (S_t, \leq_t, t, h_t, D)$*, for* $t \in S$*. The definitions will be given simultaneously for all of them.*

By an intuitionistic 'subset' of D *in* $(S, \leq, t_0, h, D)$*, we mean a function* $\mathbf{d}$ *giving for each* $t \in S$*, a subset* $\mathbf{d}(t) \subseteq D$ *such that* $t_1 \leq t_2 \Rightarrow \mathbf{d}(t_1) \subseteq \mathbf{d}(t_2)$*.*

Let $\mathbf{f}_t^n$ *be a function associating with each* $\mathbf{d}_t$ *and* $t \in S$ *a family* $\mathbf{f}_t^n(\mathbf{d}_t)$ *of level* n *models, such that* $t_1 \leq t_2 \Rightarrow \mathbf{f}_{t_1}^n(\mathbf{d}_{t_1}) \supseteq \mathbf{f}_{t_2}^n(\mathbf{d}_{t_2})$*. Then* $(\mathbf{m}_t, \mathbf{f}_t)$ *is a model of level* $n + 1$ *where* $\mathbf{d}_t = \mathbf{d} \restriction S_t$*.*

Definition 11 (Satisfaction in fibred intuitionistic models). *We define satisfaction of formulas of level* n *in models of level* $n' \geq n$ *as follows.*

Let $(\mathbf{m}_t, \mathbf{f}_t^n)$ *be a level* n *model. Let* $\alpha = \{x\}\varphi(x)$ **says** ψ *is of level* n*. We assume we know how to check satisfaction of* $\varphi(x)$ *in any of these models.*

We can assume that

$$\mathbf{d}_t = \{x \in D \mid t \vDash \varphi(x) \text{ in } (\mathbf{m}_t, \mathbf{f}_t^n)\}$$

is defined. Then $t \vDash \alpha$ *iff for all models* $\mathbf{m}'_t$ *in* $\mathbf{f}_t^n(\mathbf{d}_t)$ *we have* $\mathbf{m}'_t \vDash \psi$*.*

5 An Example

In this section we want to give an informal example of policy written in FSL. Suppose we have the following distributed policies and facts of a computer science department:

- The department of Computer Science (CS_Dep) delegates the $University$ to say who is regularly enrolled as a student

$$\begin{gathered}\forall x(University \textbf{ says } regularly\_enrolled(x) \rightarrow_{12} \\ CS\_Dep \textbf{ says } regularly\_enrolled(x))\end{gathered} \tag{3}$$

- The delegation depth between $University$ and CS_Dep must be limited to one, we have that for every principal P ($P \in D$):

$$\begin{gathered}P \Rightarrow_{regularly\_enrolled(x)} University \rightarrow \\ P = University\end{gathered} \tag{4}$$

- The CS_Dep delegates the *group* of all the members of the ICT staff to assign logins to students

$$\begin{gathered}\forall y(\{x\}ICT\_member(x) \textbf{ says } has\_login(y) \rightarrow \\ CS\_Dep \textbf{ says } has\_login(y))\end{gathered} \tag{5}$$

- The group of the ICT staff members says that z has a login if and only if one single member of the group says it

$$\begin{gathered}\{x\}ICT\_member(x) \textbf{ says } has\_login(z) \leftrightarrow \\ \exists y(ICT\_member(y) \wedge y \textbf{ says } has\_login(z))\end{gathered} \tag{6}$$

- If someone has a login and is regulary enrolled as student at the University then he can have access to his mail:

$$\begin{gathered}(CS\_Dep \textbf{ says } has\_login(x) \wedge \\ CS\_Dep \textbf{ says } regulary\_enrolled(x) \wedge \\ x \textbf{ says } can\_access\_mail(x)) \rightarrow \\ can\_access\_mail(x)\end{gathered} \tag{7}$$

- John and Adam are members of the ICT staff of che Computer Science Departement:

$$ICT\_member(John) \wedge ICT\_member(Adam) \tag{8}$$

- The University certifies that Tom is regulary enrolled:

$$University \textbf{ says } regularly\_enrolled(Tom) \tag{9}$$

[12] CS_Dep **says** ψ in formal FSL must be inteded as $\{x\}(x = CS\_Dep)$ **says** ψ

- Tom has a login which has been assigned by Adam

$$Adam\ \textbf{says}\ has\_login(Tom) \tag{10}$$

Suppose now we want to check if Tom can access his mail, so that from

$$Tom\ \textbf{says}\ can\_access\_mail(Tom) \tag{11}$$

we want, on the basis of the following knowledge base, to derive

$$can\_access\_mail(Tom) \tag{12}$$

In fact we can make the following reasoning:

- From (1) and (6) we get

$$CS\_Dep\ \textbf{says}\ regularly\_enrolled(Tom) \tag{13}$$

- From (7) and (3) we derive

$$\{x\}ICT\_member(x)\ \textbf{says}\ has\_login(Tom) \tag{14}$$

- With (2) and (11) we obtain

$$CS\_Dep\ \textbf{says}\ has\_login(Tom) \tag{15}$$

- Now with (12),(10),(8) and (4) we finally conclude

$$can\_access\_mail(Tom) \tag{16}$$

6 Conclusion and Future Works

In this paper we presented FSL, a language for access control in distributed systems. Our approach is based on fibring [1] which is a methodology to compose logics and use them within a same language. In Section 4.2 we introduced a fibred semantics to merge intuitionistic logic with modalities indexed by first-order formulas creating predicate FSL. Predicate FSL is a language which satisfies the requirements listed in Section 1, in future works we plan to first extend well known existing logics like Delegation Logic [3], SecPAL [2] and DEBAC [4] with the FSL methodology and then to translate them into predicate FSL modal logic. We are also working on providing a calculus for predicate FSL, in order to maintain the calculus tractable we plan to employ first-order modal theorem provers without resorting to second order.

References

1. D. M. Gabbay, "Fibring logics," *Oxford University Press*, 1999.
2. M. Y. Becker, C. Fournet, and A. D. Gordon, "Design and semantics of a decentralized authorization language," in *CSF*. IEEE Computer Society, 2007, pp. 3–15.
3. N. Li, B. N. Grosof, and J. Feigenbaum, "Delegation logic: A logic-based approach to distributed authorization," *ACM Trans. Inf. Syst. Secur.*, vol. 6, no. 1, pp. 128–171, 2003.
4. C. Bertolissi, M. Fernández, and S. Barker, "Dynamic event-based access control as term rewriting," in *DBSec*, ser. Lecture Notes in Computer Science, S. Barker and G.-J. Ahn, Eds., vol. 4602. Springer, 2007, pp. 195–210.
5. B. W. Lampson, M. Abadi, M. Burrows, and E. Wobber, "Authentication in distributed systems: Theory and practice," *ACM Trans. Comput. Syst.*, vol. 10, no. 4, pp. 265–310, 1992.
6. M. Abadi, M. Burrows, B. W. Lampson, and G. D. Plotkin, "A calculus for access control in distributed systems," in *CRYPTO*, ser. Lecture Notes in Computer Science, J. Feigenbaum, Ed., vol. 576. Springer, 1991, pp. 1–23.
7. M. Abadi, "Logic in access control," in *LICS*. IEEE Computer Society, 2003, pp. 228–.
8. ——, "Access control in a core calculus of dependency," *Electr. Notes Theor. Comput. Sci.*, vol. 172, pp. 5–31, 2007.
9. ——, "Variations in access control logic," in *DEON*, ser. Lecture Notes in Computer Science, R. van der Meyden and L. van der Torre, Eds., vol. 5076. Springer, 2008, pp. 96–109.
10. B. W. Lampson, "Computer security in the real world," *IEEE Computer*, vol. 37, no. 6, pp. 37–46, 2004.
11. M. D. Schroeder and J. H. Saltzer, "The protection of information in computer systems," *Procs. IEEE 63*, vol. 9, pp. 1278–1308, 1975.
12. C. Ellison, B. Frantz, B. Lampson, R. Rivest, B. Thomas, and T. Ylonen, "Spki certificate theory," *IETF RFC 2693*, 2009.
13. L. Giuri and P. Iglio, "Role templates for content-based access control," in *ACM Workshop on Role-Based Access Control*, 1997, pp. 153–159.
14. E. Lupu and M. Sloman, "Reconciling role based management and role based access control," in *ACM Workshop on Role-Based Access Control*, 1997, pp. 135–141.
15. T. Kosiyatrakul, S. Older, and S.-K. Chin, "A modal logic for role-based access control," in *MMM-ACNS*, ser. Lecture Notes in Computer Science, V. Gorodetsky, I. V. Kotenko, and V. A. Skormin, Eds., vol. 3685. Springer, 2005, pp. 179–193.

A Appendix

A.1 Axiomatisation and completeness of FSL

We prove completeness for FSL with increasing domains and for FSL with constant domains (FSL and FSL_{CD}). Well-formed formulas (wffs) are defined recursively as follows:

- Atoms of the form $P(t_1 \dots t_n)$[13] are wffs.

[13] where $t_1 \dots t_n$ are classical first-order terms.

- $\bot$ is a wff.
- If α and β are wff, then so are $(\neg\alpha), (\alpha \wedge \beta), (\alpha \vee \beta), (\alpha \to \beta), (\forall x\alpha), (\exists x\alpha)$.
- If $\varphi(x)$ and ψ are wff, the so is $\{x\}\varphi(x)\,\mathbf{says}\,\psi$.

Axiom system for predicate FSL

1. All axioms and rules for intuitionistic logic
2. Extensionality axiom:

$$\forall x(\varphi_1(x) \leftrightarrow \varphi_2(x)) \to \\ (\{x\}\varphi_1(x)\ \mathbf{says}\ \psi \leftrightarrow \{x\}\varphi_2(x)\ \mathbf{says}\ \psi)$$

3. Modality axioms:

$$\frac{\vdash \bigwedge_i \alpha \to \beta}{\vdash \bigwedge_i \{x\}\varphi\,\mathbf{says}\,\alpha_i \to \{x\}\varphi\,\mathbf{says}\,\beta}$$

4. Constant domains axioms[14]:
   (a) $\forall y\{x\}\varphi\,\mathbf{says}\,\beta(y) \to \{x\}\varphi\,\mathbf{says}\,\forall y\beta(y)$
   (b) $\forall y(\psi \vee \beta(y)) \to (\psi \to \forall y\beta(y))$
5. Additional Axioms:
   (a) $A \to \{x\}\varphi\,\mathbf{says}\,A$
   (b) *here we put all the axioms we need to craft our logic from Sections 2,3 like for instance:*

$$\forall t(\varphi(t) \to t\,\mathbf{says}\,\psi) \to \{x\}\varphi(x)\,\mathbf{says}\,\psi$$

Definitions and Lemmas

Definition 12 (Consistent and Complete Theory). *Suppose we have a theory (Δ, Θ) of sentences*[15]*.*

- *(Δ, Θ) is consistent, if we <u>do not</u> have for some $\alpha_i \in \Delta$, $\beta_j \in \Theta$*

$$\vdash \bigwedge_i \alpha_i \to \bigvee_j \beta_j$$

- *(Δ, Θ) is complete in the language with variables $\mathcal{V}$ iff for all ψ in the language, we have*

$$\psi \in \Delta \text{ or } \psi \in \Theta$$

Definition 13 (Saturated Theory). *A theory (Δ, Θ) is saturated in a language with variables $\mathcal{V}$ iff the following holds:*

1. *(Δ, Θ) is consistent*

[14] y not free in ψ or φ.

[15] intuitively, Δ is the set of formulas that are true in the model and Θ is the set of formulas that are false in the model.

2. $\exists x A(x) \in \Delta$, *then for some* $y \in \mathcal{V}$, $A(y) \in \Delta$.
3. $\forall x A(x) \notin \Delta$, *then for some* $y \in \mathcal{V}$, $A(y) \notin \Delta$
4. $A \vee B \in \Delta$ *iff* $A \in \Delta$ *or* $B \in \Delta$.
5. *If for some* $\beta_j \in \Theta$
$$\Delta \vdash A \vee \beta_j \Rightarrow A \in \Delta$$
with A *in the language with variables* $\mathcal{V}$

Definition 14 (Constant Domain Theory). *A theory* (Δ, Θ) *is said to be constant domain (CD) theory in language* $\mathcal{V}$ *iff for any* $\forall x A(x)$ *and any* $\beta_j \in \Theta$ *such that*

$$\Delta \nvdash \forall x A(x) \vee \bigvee_j \beta_j$$

then for some y

$$\Delta \nvdash A(y) \vee \bigvee_j \beta_j$$

Lemma 1. *Assume the CD axiom* $\forall x(\beta \vee A(x) \to (\beta \vee \forall x A(x)))$, *then if* (Δ, Θ) *is a consistent CD theory and* $\Delta^{'} = \Delta \cup \{\alpha_1, \ldots, \alpha_n\}$, $\Theta^{'} = \Theta \cup \{\gamma_1, \ldots, \gamma_m\}$ *and* $(\Delta^{'}, \Theta^{'})$ *is consistent then* $(\Delta^{'}, \Theta^{'})$ *is a CD theory*

Proof. Assume

$$\Delta \cup \bigwedge_i \alpha_i \nvdash (\bigvee_j \beta_j) \vee (\bigvee_j \gamma_j) \vee \forall x A(x)$$

we can assume x *not in* β_j,α_j, γ_j *hence*

$$\begin{array}{l} \Delta \nvdash \bigwedge_i \alpha_i \to \forall x(\bigvee_j \beta_j) \vee (\bigvee_j \gamma_j) \vee \forall x A(x) \\ \Delta \nvdash \forall x(\bigwedge_i \alpha_i \to (\bigvee_j \beta_j) \vee (\bigvee_j \gamma_j) \vee A(x)) \end{array}$$

hence for some y

$$\Delta \nvdash \bigwedge_i \alpha_i \to \beta \vee A(y) \vee \gamma_j$$

hence $\Delta^{'} \nvdash \beta \vee A(y) \vee \gamma_j$

Lemma 2. *Let* (Δ, Θ) *be a saturated theory. Let* $\Delta^{'}$ *be*

$$\{\psi | (\{x\}\varphi(x) \ \mathbf{says}\ \psi) \in \Delta\}$$

Assume

$$\begin{array}{l} (\{x\}\varphi(x) \ \textbf{\textit{says}}\ \beta) \in \Theta \\ \Delta^{'} \nvdash \beta \vee \forall x A(x) \end{array}$$

then for some y

$$\Theta \nvdash \beta \vee A(y)$$

Proof. *The proof is by contradiction, suppose it is not the case that*

$$\Theta \nvdash \beta \vee A(y)$$

then, for each y there exists a finite $\Delta^{'}_{y} \subseteq \Delta^{'}$ *such that*

$$\nvdash \bigwedge \Delta^{'}_{y} \rightarrow \beta \vee A(y)$$

hence, with $\alpha \in \Delta^{'}_{y}$

$$\nvdash \bigwedge \{x\}\varphi(x) \,\textbf{says}\, \alpha \rightarrow \{x\}\varphi \,\textbf{says}\, \beta \vee A(y)$$

hence, for all y

$$\{x\}\varphi \,\textbf{says}\, \beta \vee A(y) \in \Delta$$

Since Δ *is saturated we get:*

$$\forall y \{x\}\varphi \,\textbf{says}\, (\beta \vee A(y)) \in \Delta$$

hence

$$\{x\}\varphi \,\textbf{says}\, \forall y(\beta \vee A(y)) \in \Delta$$

hence

$$\forall y(\beta \vee A(y)) \in \Delta^{'}$$

but then

$$\beta \vee \forall y A(y) \in \Delta^{'}$$

which is a contradiction.

Lemma 3. *Let* (Δ, Θ) *be a consistent CD theory, then* (Δ, Θ) *can be extended to a saturated theory* $(\Delta^{'}, \Theta^{'})$ *in the same laguage with* $\Delta \subseteq \Delta^{'}$ *and* $\Theta \subseteq \Theta^{'}$

Proof. *The proof is by induction on* (Δ_n, Θ_n) *the theory, let* $\Delta_o = \Delta$ *and* $\Theta_0 = \Theta$.

Assume (Δ_n, Θ_n) *is defined,* $\Theta_n - \Theta$ *is finite and* (Δ_n, Θ_n) *is CD. Let* β_{n+1} *be the* $(n+1)$*th wff of the language. Then either* $(\Delta_n, \Theta_n \cup \beta_{n+1})$ *is consistent or is not consistent, it it is consistent let*

$$\begin{array}{l} \Delta_{n+1} = \Delta_n \\ \Theta_n + 1 = \Theta_n \cup \{\beta\} \end{array}$$

If it is inconsistent then $(\Delta_n \cup \beta, \Theta_n)$ *must be consistent so let*

$$\begin{array}{l} \Delta_{n+1} = \Delta_n \cup \{\beta\} \\ \Theta_n + 1 = \Theta_n \end{array}$$

In any case $(\Delta_{n+1}, \Theta_{n+1})$ *is CD.*

Now let $(\Delta, \Theta) = \bigcup_n (\Delta_n, \Theta_n)$, *this theory is the saturated theory.*

Definition 15. *Let* S *be the set of all complete theories in the predicate language FSL. If the logic is CD then all the theories are in the language with variables* $\mathcal{V}$*, if the logic is not CD, then assume that each theory leaves us an infinite number of variables from* $\mathcal{V}$ *not in the theory. We can write* $(\varDelta, \varTheta)$ *as* $\varDelta$ *because for a saturated theory* $(\varDelta, \varTheta)$*, we have* $\varTheta = \{\beta | \varDelta \nvdash \beta\}$*.*

Define two relations on S

1. *(set inclusion)* $\varDelta \subseteq \varDelta^{'}$
2. *For every* $\{x\}\varphi(x)$ *let* $\varDelta R_{\{x\}\varphi(x)} \varDelta^{'}$ *iff for all* ψ *such that* $\{x\}\varphi\, \textbf{says}\, \psi \in \varDelta$ *we have* $\psi \in \varDelta^{'}$*.*

Lemma 4. *Suppose* $\varDelta \nvdash \alpha \to \beta$*, then for some* $\varDelta^{'} \supseteq \varDelta$*,* $\varDelta^{'} \vdash \alpha$ *and* $\varDelta' \nvdash \beta$

Proof. *From hypothesis we have*

$$\varDelta \cup \{\alpha\} \nvdash \beta$$

and $\varDelta \cup \{\alpha\}$ *can be completed to be a saturated theory* $\varDelta^{'}$ *such that*

$$\varDelta^{'} \nvdash \beta$$

In case of logic CD, this can be done in the same language with variables $\mathcal{V}$*. If the logic is not CD, then since there is an infinite number of variables not in* $\varDelta$*,* $\varDelta^{'}$ *can use some of them, still leaving infinitely out of* $\varDelta$

Lemma 5. *Assume* $\varDelta \nvdash \forall x \varphi(x)$*, if the logic is not CD, then for some* u *not in the language od* $\varDelta$*, we have* $\varDelta \nvdash \varphi(x)$*.* $\varDelta$ *can be extended in a saturated* $\varDelta^{'}$ *by adding the variable* u *and more variables such that* $\varDelta^{'} \nvdash \varphi(u)$*, and still infinitely numbers of variables are not in* $\varDelta^{'}$*. If the logic is CD, such a* u *is in the logic in* $\varDelta$ *and* $(\varDelta, \{\varphi(u)\})$ *can be extended to a complete and saturated theory in the same language.*

Lemma 6. *Let* $(\varDelta, \varTheta)$ *be complete and saturated. Assume* $\{x\}\varphi\, \textbf{says}\, \psi$ *is not in* $\varTheta$*. Then*

$$\varDelta_0 = \{\alpha | \{x\}\varphi(x)\, \textbf{says}\, \alpha \in \varDelta\}$$

does not prove ψ*, otherwise*

$$\vdash \bigwedge \alpha_j \to \psi$$

hence

$$\vdash \bigwedge_j \{x\}\varphi(x)\, \textbf{says}\, \alpha_i \to \{x\}\varphi(x)\, \textbf{says}\, \psi$$

hence

$$\{x\}\varphi(x)\, \textbf{says}\, \in \varDelta$$

Since $\varDelta_0$ *does not prove* ψ*, and* $(\varDelta_0, \{\psi\})$ *is consistent, we can extend* $\varDelta_0$ *to a saturated theory* $(\varDelta^{'}, \varTheta^{'})$*. In case the logic is CD,* $(\varDelta^{'}, \varTheta^{'})$ *will be in the same language. Otherwise we use more variables.*

Lemma 7. *Properties of the model $(S, \subseteq, R_{\{x\}\varphi})$:*

1. *$\Delta_1 \subseteq \Delta_2$ and $\Delta_2 R_{\{x\}\varphi}\Theta$ then $\Delta_1 R_{\{x\}\varphi}\Theta$*

   Proof. *$\Delta_2 R_{\{x\}\varphi}\Theta$ means for every $\{x\}\varphi$ **says** $\psi \in \Delta_2$ we have $\psi \in \Theta$. Since $\Delta_1 \subseteq \Delta_2$ we have for every $\{x\}\varphi$ **says** $\psi \in \Delta$ we have $\psi \in \Theta$.*

2. *If we add the axiom $\forall x(\varphi(x) \leftrightarrow \varphi^{'}(x)) \to (\{x\}\varphi$ **says** $\psi \leftrightarrow \{x\}\varphi^{'}$ **says** $\psi)$ we get the condition*

   $$\Delta \vdash \forall x(\varphi(x) \leftrightarrow \varphi^{'}(x))$$

   implies for all Θ

   $$\Delta R_{\{x\}\varphi}\Theta \leftrightarrow \Delta R_{\{x\}\varphi'}\Theta$$

Definition 16 (Construction of the model). *Take $(S, \subseteq, R_{\{x\}\varphi(x)})$ as defined above. For atomic $P(x_1, \ldots, x_n)$ and $\Delta \in S$, let*

$$\Delta \models P \text{ iff } P \in \Delta$$

The domain of Δ is defined by the variables of Δ. If the logic is CD all Δ will have variables $\mathcal{V}$ as domain, otherwise we will have variable domains.

Lemma 8. *For any ψ, Δ*

$$\Delta \models \psi \text{ iff } \psi \in \Delta$$

Proof. *Proof by taking in exam ”$\rightarrow$” and ”**says**”.*

How Do Agents Comply with Norms?

Guido Governatori[1] and Antonino Rotolo[2]

[1] NICTA, Queensland Research Laboratory
guido.governatori@nicta.com.au
[2] CIRSFID and Law Faculty, University of Bologna
antonino.rotolo@unibo.it

Abstract. The import of the notion of institution in the design of MASs requires to develop formal and efficient methods for modeling the interaction between agents' behaviour and normative systems. This paper discusses how to check whether agents' behaviour is compliant with the rules regulating them. The key point of our approach is that compliance is a relationship between two sets of specifications: the specifications for executing a process and the specifications regulating it. We propose a logic-based formalism for describing both the semantics of normative specifications and the semantics of compliance checking procedures.

1 Introduction

Recent developments in MAS have pointed out that normative concepts can play a crucial role in modeling agents' interaction [29]. In fact, while the main objective is to design systems of autonomous agents, it is likewise important that agent systems may exhibit global desirable properties. Like in human societies, such properties are ensured if the interaction of artificial agents, too, adopts institutional and organizational models whose goal is to regiment agents' behaviour through normative systems in supporting coordination, cooperation and decision-making. However, to keep agents autonomous it is often suggested that norms should not simply work as hard constraints, but rather as soft constraints [6]. In this sense, norms should not limit in advance agents' behaviour, but would instead provide standards which can be violated, even though any violations should result in sanctions or other normative effects applying to non-compliant agents.

If normative systems for MAS are designed as mentioned above, it is of paramount importance to develop mechanisms to characterizing and detecting agents' *norm compliance*. To our knowledge, no systematic investigation has been devoted so far to this research issue in MAS theory, whereas its importance has increased over the last few years in other related fields such as in business modeling. In this perspective, compliance is essentially ensuring that business processes, operations and practise are in accordance with a prescribed and/or agreed set of norms. Compliance is often used to denote adherence of one set of rules (we refer to them as 'source rules' hereafter) against other set of rules (we refer to them as 'target rules' hereafter).

In this paper we apply this interpretation of compliance to discuss adherence or consistence of a set of rules specifying a process against a set of "normative" rules regulating it. Of course, agents' compliance could be tested by directly focusing on plan

Dagstuhl Seminar Proceedings 09121
Normative Multi-Agent Systems
http://drops.dagstuhl.de/opus/volltexte/2009/1909

design and execution. The choice of working on processes is motivated by two reasons. First, modelling agents' behaviour in terms of processes has been proven useful in developing agent-oriented systems for business management (for a recent proposal see, e.g., [5]). The correspondence of business processes and agent plans makes business services flexible and adaptable. Second, while it is far from obvious that complex plan's actions can be always viewed as processes (for the pros and cons of this view, see [9]), in institutional settings agents usually instantiate roles, which consist of a specification of an agent's internal and external behavior. In this sense, taking roles as specific processes (or procedures) allows for obtaining a flexible team agent structure [28]. Under this working hypothesis, the problem of norm compliance can be framed as the relationship between the specifications for process execution and those regulating it.

Process specifications describe how a process is executed while norms state what can be done and what cannot be done by a process. The problem is how to align the language to specify the activities to be performed to complete a process and the conditions set up by the norms relevant for the process. The solution of such a problem is not trivial matter. The detection of violations and the design of agents' compliance amount to relatively affordable operations when we have to check whether processes are compliant with respect to simple normative systems. But things are tremendously harder when we deal with processes to be tested against realistic, large and articulated systems of norms.

What do we mean by a "complex" normative system? Among other things, the complexities of normative systems reside in the fact that they regulate agents's behaviour by usually specifying actions to be taken in case of breaches of some of the norms, actions which can vary from (pecuniary) penalties to the termination of an interaction itself. These constructions, i.e., obligations in force after some other obligations have been violated, are known in the deontic literature as contrary-to-duty obligations (CTDs) or reparational obligations (because they are meant to 'repair' or 'compensate' violations of primary obligations [7]). Thus a CTD is a conditional obligation arising in response to a violation, where a violation is signalled by an unfulfilled obligation. These constructions identify situations that are not ideal for the interaction but still acceptable. The ability to deal with violations and the reparational obligations generated from them is an essential requirement for agents where, due to the nature of the environment where they are deployed, some failures can occur, but it does not necessarily mean that the whole interaction has to fail. However, the main problem with these constructions is that they can give rise to very complex rule dependencies, because we can have that the violation of a single rule can activate other (reparational) rules, which in turn, in case of their violation, refer to other rules, and so forth.

In this paper, we take inspiration from an approach originally designed for modeling business process compliance[3]. This approach is based on (semantic) annotations, where the annotations are written in the formal language chosen to represent the normative specifications. The idea is that processes are annotated and the annotations provide the conditions a process has to comply with. Annotations can be at different levels; for example we can annotate a full process or a single task in a process. In addition we can have different types of annotation. Annotations can range from the full set of rules

[3] For a comprehensive exposition of compliance for business process models, see [21, 27]

(norms) specific to a process or a single task to simple semantic annotation corresponding to one effect of a particular task, e.g., after the successful execution of task A in a process B the value of the environment variable C is D.

The layout of the paper is as follows. Section 2 provides a reasoning mechanism to deal with reparational constructions and to reframe the normative system in such a way as it is possible to detect agent compliance. Section 3 briefly outlines how to represent processes and to annotate them. Section 4 provides a semantics of compliance checking procedures on account of what proposed in the previous sections. A section on related work ends the paper.

2 Normative Constraints for MAS

The expression of violation conditions and the reparation obligations is an important requirement for formalising norms, design subsequent processes to minimise or deal with such violations and also to determine the compliance of a process with the relevant norms. The violation expression consists of the primary obligation, its violation conditions, an obligation generated upon the violation condition occurs, and this can recursively be iterated, until the final condition is reached. This final condition is one which cannot be violated and this it is to be a permission. We introduce the non-boolean connective $\otimes$, whose interpretation is such that $OA \otimes OB$ is read as "OB is the reparation of the violation of OA". In other words the interpretation of $OA \otimes OB$, is that A is obligatory, but if the obligation OA is not fulfilled (i.e., when $\neg A$ is the case), then the obligation OB is activated and becomes in force until it is satisfied or violated. In the latter case a new obligation may be activated, followed by others in chain, as appropriate.

2.1 Process Compliance Language (PCL)

We now provide a formal account of the idea presented above. Our formalism, called Process Compliance Language (PCL), is a combination of an efficient non-monotonic formalism (defeasible logic [3,4]) and a deontic logic of violations [18]. This particular combination allows us to represent exceptions as well as the ability to capture violations and the obligations resulting from the violations; in addition our framework has good computational properties: the extension of a theory (i.e., the set of conclusions/normative positions following from a set of facts) can be computed in time linear to the size of the theory.

The ability to handle violation is very important for compliance of agents' processes. Often agents operate in dynamic and somehow unpredictable environments. As a consequence in some cases, maybe due to external circumstances, it is not possible to operate in the way specified by the norms, but the norms prescribe how to recover from the resulting violations. In other cases, the prescribed behaviours are subject to exceptions. Finally, in other cases, one might not have a complete description of the environment. Accordingly the process has to operate based on the available input (this is typically the case of the *due diligence* prescription), but if more information were available, then the task to be performed could be a different one. A conceptually sound formalisation of norms (for assessing the compliance of a process) should take into account all the aspects mentioned above. PCL is sound in this respect given the combinations of the deontic component (able to represent the fundamental normative positions and chains of violations/reparations) and the defeasible component that takes care of the issue about partial information and possibly conflicting prescriptions.

Our formal language consists of the following set of atomic symbols: a numerable set of propositional letters $p, q, r, \ldots$, intended to represent the state variables and the tasks of a process. Formulas of the logic are constructed using the deontic operators O (for obligation), P (for permission), negation $\neg$ and the non-boolean connective $\otimes$ (for the Contrary-To-Duty (CTD) operator). The formulas of PCL will be constructed in two steps according to the following formation rules:

- every propositional letter is a literal;
- the negation of a literal is a literal;
- if X is a deontic operator and l is a literal then Xl and $\neg Xl$ are deontic literals.

After we have defined the notions of literal and deontic literal we can use the following set of formation rules to introduce $\otimes$-expressions, i.e., the formulas used to encode chains of obligations and violations.

- every deontic literal is an $\otimes$-expression;
- if $Ol_1, \ldots, Ol_n$ are deontic literals and l_{n+1} is a literal, then $Ol_1 \otimes \ldots \otimes Ol_n$ and $Ol_1 \otimes \ldots \otimes Ol_n \otimes Pl_{n+1}$ are $\otimes$-expressions.

The connective $\otimes$ permits combining primary and CTD obligations into unique regulations. The meaning of an expression like $O_sA \otimes O_sB \otimes O_sC$ is that the primary obligation for agent s is A, but if A is not done, then s has the obligation to do B. But if event B fails to be realised, then s has the obligation to do C. Thus B is the reparation of the violation of the obligation O_sA ($\neg A$ holds). Similarly C is the reparation of the obligation O_sB, which is in force when the violation of A occurs.

The formation rules for $\otimes$-expressions allow a permission to occur only at the end of such expressions. This is due to the fact that a permission can be used as a reparation of a violation, but it is not possible to violate a permission, thus it makes no sense to have reparations to permissions.

Each norm is represented by a rule in PCL, where a rule is an expression $r : A_1, \ldots, A_n \Rightarrow C$, where r is the name/id of the norm, $A_1, \ldots, A_n$, the *antecedent* of the rule, is the set of the premises of the rule (alternatively it can be understood as the conjunction of all the literals in it) and C is the conclusion of the rule. Each A_i is either a literal or a deontic literal and C is an $\otimes$-expression.

The meaning of a rule is that the normative position (obligation, permission, prohibition) represented by the conclusion of the rule is in force when all the premises of the rule hold.

PCL is equipped with a superiority relation (a binary relation) over the rule set. The superiority relation ($\prec$) determines the relative strength of two rules, and it is used when rules have potentially conflicting conclusions. For example given the rules $r_1 : A \Rightarrow B \otimes C$ and $r_2 : D \Rightarrow \neg C$. $r_1 \prec r_2$ means that rule r_1 prevails over rule r_2 in situations where both fire and they are in conflict (i.e., rule r_1 fires for the secondary obligation C). For example let us consider the following two contract rules[4]:

$$r : PremiumCustomer \Rightarrow O_s Discount$$
$$r' : SpecialOrder \Rightarrow O_s \neg Discount$$

saying that Premium Customers are entitled to a discount (r), but there is no discount for goods bought with a special order (r'). Is a Premium customer entitled to a discount when she places a special order? If we only have the two rules above there is no way to solve the conflict just using the contract and there is the need of a domain expert to advise the knowledge engineer about what to do in such case. The logic can only point out that there is a conflict in the contract. On the other hand, if we have an additional provision

$$r'' : PremiumCustomer, \neg Discount \Rightarrow O_s Rebate$$

specifying that if for some reasons a premium customer did not receive a discount then the customer is entitled to a rebate on the next order, then it is possible to solve the conflict, because the contract allows a violation of rule r to be amended by r'', using the merging mechanism we analyse in Section 2.2.

2.2 Normal Forms

We introduce transformations of an PCL representation of a normative system to produce a normal form of the same (NPCL). A normal form is a representation of a normative system based on an PCL specification containing all conditions that can be generated/derived from the given PCL specification. The purpose of a normal form is to "clean up" the PCL representation of a normative system, that is to identify formal loopholes, deadlocks and inconsistencies in it, and to make hidden conditions explicit.

In the rest of this section we introduce the procedures to generate normal forms. First (Section 2.2) we describe a mechanism to derive new conditions by merging together existing normative clauses. In particular we link an obligation and the obligations triggered in response to violations of the obligation. Then, in Section 2.2, we examine the problem of redundancies, and we give a condition to identify and remove redundancies from the formal normative specification.

Merging Norms One of the features of the logic of violations is to take two rules, or norms, and merge them into a new clause. In what follows we will first examine some common patterns of this kind of construction and then we will show how to generalise them.

Consider a norm like (Γ and Δ are sets of premises)

$$\Gamma \Rightarrow O_s A.$$

[4] In what follows we will use O_S and P_S for the obligation and permission operators relative to the *Supplier*, and O_P and P_P for the *Purchaser*. O_s and P_s will be used for a generic subject.

Given an obligation like this, if we have that the violation of $O_s A$ is part of the premises of another norm, for example,

$$\Delta, \neg A \Rightarrow O_{s'} C,$$

then the latter must be a good candidate as reparational obligation of the former. This idea is formalised as follows:

$$\frac{\Gamma \Rightarrow O_s A \qquad \Delta, \neg A \Rightarrow O_{s'} C}{\Gamma, \Delta \Rightarrow O_s A \otimes O_{s'} C}$$

This reads as follows: given two policies such that one is a conditional obligation ($\Gamma \Rightarrow O_s A$) and the antecedent of second contains the negation of the propositional content of the consequent of the first ($\Delta, \neg A \Rightarrow O_{s'} C$), then the latter is a reparational obligation of the former. Their reciprocal interplay makes them two related norms so that they cannot be viewed anymore as independent obligations. Therefore we can combine them to obtain an expression (i.e., $\Gamma, \Delta \Rightarrow O_s A \otimes O_{s'} C$) that exhibits the *explicit reparational obligation* of the second norm with respect to the first. Notice that the subject of the primary obligation and the subject of its reparation can be different, even if very often they are the same.

Suppose we have the following rules

$$r : \mathit{Invoice} \Rightarrow O_P \mathit{PayWithin7Days}$$
$$r' : \neg \mathit{PayWithin7Days} \Rightarrow O_P \mathit{PayWithInterest}.$$

From these we obtain

$$r'' : \mathit{Invoice} \Rightarrow O_P \mathit{PayWithin7Days} \otimes O_P \mathit{PayWithInterest}.$$

We can also generate chains of CTDs in order to deal iteratively with violations of reparational obligations. The following case is just an example of this process.

$$\frac{\Gamma \Rightarrow O_s A \otimes O_s B \qquad \neg A, \neg B \Rightarrow O_s C}{\Gamma \Rightarrow O_s A \otimes O_s B \otimes O_s C}$$

For example, from the rules

$$r : \mathit{Invoice} \Rightarrow O_S \mathit{QualityOfService} \otimes O_S \mathit{Replace3days}$$
$$r' : \neg \mathit{QualityOfService}, \neg \mathit{Replace3days} \Rightarrow O_S \mathit{Refund\&Penalty}$$

we derive the new rule

$$\begin{aligned} r'' :& \mathit{Invoice} \Rightarrow O_S \mathit{QualityOfService} \otimes \\ & O_S \mathit{Replace3days} \otimes O_S \mathit{Refund\&Penalty}. \end{aligned}$$

The above patterns are just special instances of the general mechanism described in details in [18, 15].

Removing Redundancies Given the structure of the inference mechanism it is possible to combine rules in slightly different ways, and in some cases the meaning of the rules resulting from such operations is already covered by other rules. In other cases the rules resulting from the merging operation are generalisations of the rules used to produce them, consequently, the original rules are no longer needed in the specifications. To deal with this issue we introduce the notion of subsumption between rules. Intuitively a rule subsumes a second rule when the behaviour of the second rule is implied by the first rule.

We first introduce the idea with the help of some examples and then we show how to give a formal definition of the notion of subsumption appropriate for PCL.

Let us consider the rules

$$r : \mathit{Service} \Rightarrow O_S\mathit{QualityOfService} \otimes O_S\mathit{Replace3days} \otimes O_S\mathit{Refund\&Penalty},$$
$$r' : \mathit{Service} \Rightarrow O_S\mathit{QualityOfService} \otimes O_S\mathit{Replace3days}.$$

The first rule, r, subsumes the second r'. Both rules state that after the supplier has provided the service she has the obligation to provide the service according to the published standards, if she violates such an obligation, then the violation of *QualityOfService* can be repaired by replacing the faulty service within three days ($O_S\mathit{Replace3days}$). In other words $O_S\mathit{Replace3days}$ is a secondary obligation arising from the violation of the primary obligation $O_S\mathit{QualityOfService}$. In addition r prescribes that the violation of the secondary obligation $O_S\mathit{Replace3days}$ can be repaired by $O_S\mathit{Refund\&Penalty}$, i.e., the seller has to refund the buyer and in addition she has to pay a penalty.

The conditions of a normative system cannot be taken in isolation insofar as they exist in it. Consequently the whole normative system determines the meaning of each single clause (norm). In agreement with this holistic view of norms we have that the normative content of r' is included in that of r. Accordingly r' does not add any new piece of information, it is redundant and can be dispensed from the explicit formulation of the norms.

Another common case is exemplified by the rules:

$$r : \mathit{Invoice} \Rightarrow O_P\mathit{PayWithin7Days} \otimes O_P\mathit{PayWithInterest}$$
$$r' : \mathit{Invoice}, \neg\mathit{PayWithin7Days} \Rightarrow O_P\mathit{PayWithInterest}.$$

The first rule says that after the seller sends the invoice the buyer has one week to pay, otherwise the buyer has to pay the principal plus the interest. Thus we have the primary obligation $O_P\mathit{PayWithin7Days}$, whose violation is repaired by the secondary obligation $O_P\mathit{PayWithInterest}$, while, according to the second rule, given the same set of circumstances *Invoice* and $\neg\mathit{PayWithin7Days}$ we have the primary obligation $O_P\mathit{PayWithInterest}$. However, the primary obligation of r' obtains when we have a violation of the primary obligation of r. Thus the condition of applicability of the second rule includes that of the first rule, which then is more general than the second and we can discard r' from the formal representation of the specifications.

The intuitions we have just exemplified is captured by the following definition.

Definition 1. *Let* $r_1 : \Gamma \Rightarrow A \otimes B \otimes C$ *and* $r_2 : \Delta \Rightarrow D$ *be two rules, where* $A = \bigotimes_{i=1}^{m} A_i$, $B = \bigotimes_{i=1}^{n} B_i$ *and* $C = \bigotimes_{i=1}^{p} C_i$. *Then* r_1 subsumes r_2 *iff (1)* $\Gamma = \Delta$ *and* $D = A$*; or (2)* $\Gamma \cup \{\neg A_1, \dots, \neg A_m\} = \Delta$ *and* $D = B$*; or (3)* $\Gamma \cup \{\neg B_1, \dots, \neg B_n\} = \Delta$ *and* $D = A \otimes \bigotimes_{i=0}^{k \leq p} C_i$.

The intuitions is that the normative content of r_2 is fully included in r_1. Thus r_2 does not add anything new to the system and it can be safely discarded.

Conflicts often arise in normative systems. What we have to determine is whether we have genuine conflicts, i.e., the norms are in some way flawed or whether we have *prima-facie* conflicts. A prima-facie conflict is an apparent conflict that can be resolved when we consider it in the context where it occurs and if we add more information the conflict disappears.

The following rule is devised for making explicit conflicting norms (contradictory norms) within the system:

$$\frac{\Gamma \Rightarrow A \qquad \Delta \Rightarrow \neg A}{\Gamma, \Delta \Rightarrow \bot} \tag{1}$$

where

1. there is no rule $\Gamma' \Rightarrow X$ such that either $\neg A \in \Gamma'$ or $X = A \otimes B$;
2. there is no conditional rules $\Delta' \Rightarrow X$ such that either $A \in \Delta'$ or $X = \neg A \otimes B$;
3. for any formula B, $\{B, \neg B\} \not\subseteq \Gamma \cup \Delta$.

The meaning of these three conditions is that given two rules, we have a conflict if the normative content of the two rules is opposite, such that none of them can be repaired, and the states of affairs/preconditions they require are consistent.

Once conflicts have been detected there are several ways to deal with them. The first thing to do is to determine whether we have a *prima-facie* conflict or a genuine conflict. As we have seen we have a conflict when we have two rules with opposite conclusions. Thus a possible way to solve the conflict is to create a superiority relation over the rules and to use it do “defeat” the weaker rule. In Section 2.3 we will examine how to reason with norms, and we will see how to use the superiority relation to solve conflicts.

Normalisation Process We now describe how to use the machinery presented in Section 2.2 and Section 2.2 to obtain PCL normal forms. The PCL normal form of a normative system provides a logical representation of normative specifications in a format that can be used to check the compliance of a process. This consists of the following steps:

1. Starting from a formal representation of the explicit clauses of a set of normative specifications we generate all the implicit conditions that can be derived from the normative system by applying the merging mechanism of PCL.
2. We can clean the resulting representation by throwing away all redundant rules according to the notion of subsumption.
3. Finally we use the conflict identification rule to label and detect conflicts.

In general the process at step 2 must be done several times in the appropriate order as described above. The normal form of a set of rules in PCL is the fixed-point of the above constructions. A normative system contains only finitely many rules and each rule has finitely many elements. In addition it is possible to show that the operation on which the construction is defined is monotonic [18], thus according to standard set theory results the fixed-point exists and it is unique. However, we have to be careful since merging first and doing subsumption after produces different results from the opposite order (i.e., subsumption first and merging after), or by interleaving the two operations.

2.3 Reasoning with Norms

In the previous section we have examined the mechanism to obtain a set of rules covering all possible (explicit) norms for obligations, permissions and prohibitions that can arise from an initial set of norms. In this section we focus on the issue of how to determine what obligations are in force for a specific situation, thus taking the well known distinction between schema and instance. The previous section defines the procedure to obtain the full (normalised) schema corresponding to a normative system. Here we study how to get the normative positions active for a specific instance of a process. The reasoning mechanism of PCL is an extension of Defeasible Logic.

Defeasible logic [24] is a simple and efficient rule based non-monotonic formalism. Over the year the logic has been developed and extended, and several variants have been proposed to model different aspects of normative reasoning and encompassed other formalisms to for normative reasoning.

The main intuition of the logic is to be able to derive "plausible" conclusions from partial and sometimes conflicting information. Conclusions are *tentative* conclusions, in the sense that a conclusion can be withdrawn when we have new pieces of information.

The knowledge in a Defeasible Theory is organised in *facts* and *rules* and *superiority relation*. Facts are indisputable statements. Defeasible rules are rules that can be defeated by contrary evidence. The superiority relation is a binary relation defined over the set of rules. The superiority relation determines the relative strength of two (conflicting) rules. The meaning of a defeasible rule, like $A_1, \ldots, A_n \Rightarrow C$, is that normally we are allowed to derive C given $A_1, \ldots, A_n$, unless we have some reasons to support the opposite conclusion (i.e., we have a rule like $B_1, \ldots, B_m \Rightarrow \neg C$).

Defeasible Logic is a "skeptical" non-monotonic logic, meaning that it does not support contradictory conclusions. Instead Defeasible Logic seeks to resolve conflicts. In cases where there is some support for concluding A but also support for concluding $\neg A$, Defeasible Logic does not conclude either of them (thus the name skeptical). If the support for A has priority over the support for $\neg A$ then A is concluded.

A defeasible conclusion is a tentative conclusion that might be withdrawn by new pieces of information, or in other terms it is the 'best' conclusion we can reach with the given information. In addition the logic is able to tell whether a conclusion is or is not provable. Thus it is possible to have the following types of conclusions: (a) Positive defeasible conclusions, meaning that the conclusions can be defeasible proved; (b) Negative defeasible conclusions, meaning that one can show that the conclusion is not even defeasibly provable. A defeasible conclusion A can be derived if there is a rule whose conclusion is A, whose prerequisites (antecedent) have either already been proved or given in the case at hand (i.e., facts), and any stronger rule whose conclusion is $\neg A$ (the negation of A) has prerequisites that fail to be derived. In other words, a conclusion A is (defeasibly) derivable when: (1) A is a fact; or (2) there is an applicable defeasible rule for A, and either (2.1) all the rules for $\neg A$ are discarded (i.e., not applicable) or (2.2) every applicable rule for $\neg A$ is weaker than an applicable strict or defeasible rule for A. A rule is applicable if all elements in the body of the rule are derivable (i.e., all the premises are positively provable), and a rule is discarded if at least one of the elements of the body is not provable (or it is a negative defeasible conclusion).

Defeasible Logic at Work We illustrate the inferential mechanism of Defeasible Logic with the help of an example. Let us assume we have a theory containing the following rules:

$$r_1 : \mathit{PremiumCustomer}(X) \Rightarrow \mathit{Discount}(X)$$
$$r_2 : \mathit{SpecialOrder}(X) \Rightarrow \neg \mathit{Discount}(X)$$
$$r_3 : \mathit{Promotion}(X) \Rightarrow \neg \mathit{Discount}(X)$$

where the superiority relation is thus defined: $r_1 \prec r_3$ and $r_2 \prec r_1$. The theory states that services in promotion are not discounted, and so are special orders with the exception of special orders placed by premium customers, who are normally entitled to a discount.

In a scenario where all we have is that we received a special order, then we can conclude that the price has to be calculated without a discount since rule r_1 is not applicable (we do not know whether the customer is a premium customer or not). In case the special order is received from a special customer for a service not in promotion, we can derive that the customer is entitled to a discount. Indeed rule r_1 is now applicable and it is stronger than rule r_2, and r_3, which is stronger than r_1, is not applicable (i.e., the service is not in promotion).

Adding Reparation Chains PCL is an extension of defeasible logic with the reparation operator ($\otimes$). Accordingly the reasoning mechanism to derive conclusion is an extension of that for defeasible logic. In defeasible logic the conclusions of a rule is a single literal and not a reparation chain. Thus the condition that *OA* appears in the conclusion of a rule means in defeasible logic that *OA* is the conclusions of the rule. For PCL have to extend the notion to accommodate reparation chain. The required change is that to prove *OA*, we have to consider all rules with a reparation chain for *OA*, where for all elements before *OA* in the chain, the negation of the element is already provable. Thus to prove *OA* given a rule $P_1, \dots, P_n \Rightarrow OC_1 \otimes \cdots \otimes OC_m \otimes OA \otimes OD_1 \otimes \cdots \otimes OD_k$, we have that $P_1, \dots, P_n$ must be all provable, and so must be $\neg C_1, \dots, \neg C_m$ [15].

3 Process Modelling

A business process model (BPM) describes the tasks to be executed (and the order in which they are executed) to fulfill some objectives of a business. BPMs aim to automate and optimise business procedures and are typically given in graphical languages. A language for BPM usually has two main elements: tasks and connectors. Tasks correspond to activities to be performed by actors (either human or artificial) and connectors describe the relationships between tasks: a minimal set of connectors consists of sequence (a task is performed after another task), parallel –and-split and and-join– (tasks are to be executed in parallel), and choice –(x)or-split and (x)or-join– (at least (most) one task in a set of task must be executed).

3.1 Execution Semantics

The basic execution semantics of the control flow aspect of a business process model is defined using token-passing mechanisms, as in Petri Nets. The definitions used here

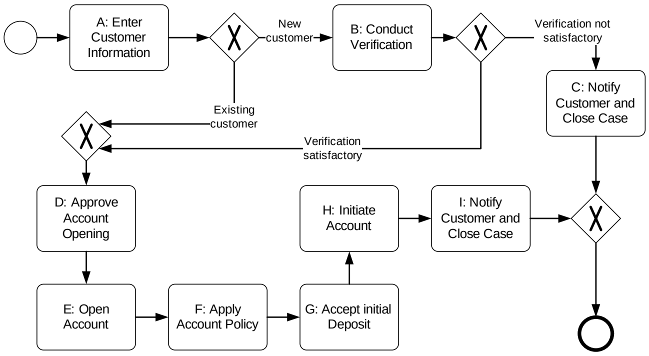

A: Enter Customer Information
New customer
B: Conduct Verification
Verification not satisfactory
C: Notify Customer and Close Case
Existing customer
Verification satisfactory
D: Approve Account Opening
H: Initiate Account
I: Notify Customer and Close Case
E: Open Account
F: Apply Account Policy
G: Accept initial Deposit

Fig. 1. Example account opening process in private banking

extend the execution semantics for process models given by [30] with semantic annotations in the form of effects and their meaning.

A process model is seen as a graph with nodes of various types –a single start and end node, task nodes, XOR split/join nodes, and parallel split/join nodes– and directed edges (expressing sequentiality in execution). The number of incoming (outgoing) edges are restricted as follows: start node 0 (1), end node 1 (0), task node 1 (1), split node 1 (>1), and join node >1 (1). The location of all tokens, referred to as a *marking*, manifests the state of a process execution. An execution of the process starts with a token on the outgoing edge of the start node and no other tokens in the process, and ends with one token on the incoming edge of the end node and no tokens elsewhere. Task nodes are executed when a token on the incoming link is consumed and a token on the outgoing link is produced. The execution of a XOR (Parallel) split node consumes the token on its incoming edge and produces a token on one (all) of its outgoing edges, whereas a XOR (Parallel) join node consumes a token on one (all) of its incoming edges and produces a token on its outgoing edge.

3.2 Annotation of Processes

A process model is then extended with a set of annotations, where the annotations describe (i) the artifacts or effects of executing and (ii) the rules describing the obligations (and other normative positions) relevant for the process.

As for the semantic annotations, the vocabulary is presented as a set of predicates P. There is a set of process variables (x and y in Table 1), over which logical statements can be made, in the form of literals involving these variables. The task nodes can be annotated using *effects* (also referred to as *postconditions*) which are conjunctions of literals using the process variables. The meaning is that, if executed, a task changes the state of the world according to its effect: every literal mentioned by the effect is true in the resulting world; if a literal l was true before, and is not contradicted by the effect, then it is still true (i.e., the world does not change of its own accord).

The obligations for this example are motivated by the following scenario: A new legislative framework has recently been put in place in Australia for anti-money laundering. The first phase of reforms for the *Anti-Money Laundering and Counter-*

Terrorism Financing Act 2006 (AML/CTF), covers the financial sector including banks, credit unions, building societies and trustees and extends to casinos, wagering service providers and bullion dealers. The act namely AML/CTF imposes a number of obligations, which include: customer due diligence (identification, verification of identity and ongoing monitoring of transactions); reporting (suspicious matters, threshold transactions and international funds transfer instructions); and record keeping. Table 1 shows the semantic effect annotations of the process activities.

Task	Semantic Annotation	Task	Semantic Annotation
A	$newCustomer(x)$	B	$checkIdentity(x)$
C	$checkIdentity(x)$, $recordIdentity(x)$	D	$accountApproved(x)$
E	$owner(x,y)$, $account(y)$	F	$accountType(y,type)$
G	$positiveBalance(y)$	H	$\neg positiveBalance(y)$
I	$accountActive(y)$	J	$notify(x,y)$

Table 1. Annotations for the process in Fig 1.

Here we give the norms governing this particular class of processes.

- All new customers must be scanned against provided databases for identity checks.

$$r_1 : newCustomer(x) \Rightarrow OcheckIdentity(x)$$

The meaning of the predicate $newCustomer(x)$ is that the input data with $Id = x$ is a new customer, for which we have the obligation to check the provided data against provided databases $checkIdentity(x)$. The obligation resulting from this rule is a non-persistent obligation, i.e. as soon as a check has been performed, the obligation is no longer in force.

- Retain history of identity checks performed.

$$r_2 : checkIdentity(x) \Rightarrow OrecordIdentity(x)$$

This rule establishes that there is a permanent obligation to keep record of the identity corresponding to the (new) customer identified by x. In addition this obligation is not fulfilled by the achievement of the activity (for example, by storing it in a database). We have a violation of the condition, if for example, the record x is deleted from the database.

- Accounts must maintain a positive balance, unless approved by a bank manager, or for VIP customers.

$$r_3 : account(y) \Rightarrow OpositiveBalance(y) \otimes OapproveManager(y)$$

The primary obligation is that each account has to maintain a positive balance *positiveBalance*; if this condition is violated (for any reason the account is not positive), then we still are in an acceptable situation if a bank manager approve the account not to be positive. In this case the obligation of approving persists until a manager approves the situation; after the approval the obligation is no longer in force.

$$r_4 \ : \ \mathit{account}(x), \mathit{owner}(x,y), \mathit{accountType}(x, VIP) \ \Rightarrow \ P\neg \mathit{positiveBalance}(x)$$

This rule creates an exception to rule r_3. Accounts of type VIP are allowed to have a non positive balance and no approval is required for this type of accounts (this is achieved by imposing that rule r_4 is stronger than rule r_3, $r_4 \prec r_3$).

4 Compliance Checking

Our aim in the compliance checking is to figure out (a) which obligations will definitely appear when executing the process, and (b) which of those obligations may not be fulfilled. In a way, PCL constraint expressions for a normative system define a behavioural and state space which can be used to analyse how well different behaviour execution paths of a process comply with the PCL constraints. Our aim is to use this analysis as a basis for deciding whether execution paths of a process are compliant with the PCL and thus with the normative system modelled by the PCL specifications. To this end we use the following procedure:

1. We traverse the graph describing the process and we identify the sets of effects (sets of literals) for all the tasks (nodes) in the process according to the execution semantics outlined in Section 3.1.
2. For each task we use the set of effects for that particular task to determine the normative positions (obligations, permissions, prohibitions) triggered by the execution of the task. This means that effects of a task are used as a set of facts, and we compute the conclusions of the defeasible theory resulting from the effects and the PCL rules annotating the process (see Sections 2 and 3.2). In the same way we accumulate effects, we also accumulate (undischarged) obligations from one task in the process to the task following it in the process.
3. For each task we compare the effects of the tasks and the obligations accumulated up to the task. If an obligation is fulfilled by a task, we discharge the obligation, if it is violated we signal this violation. Finally if an obligation is not fulfilled nor violated, we keep the obligation in the stack of obligations and propagate the obligation to the successive tasks.

Here, we assume that the obligations derived from a task should be fulfilled in the remaining of the process. Variations of this schema are possible: for example, one could stipulate that the obligations derived from a task should be fulfilled by the tasks immediately after the task. In another approach one could use a schema where for each task one has both preconditions and effects. Then the obligations derived from the preconditions must be fulfilled by the current task (i.e., the obligations must be fulfilled by the effects of the task), and the obligations derived from the effects are as in our basic schema.

4.1 From Tasks to Obligations

The second step to check process compliance is to determine the obligations derived by the effects of a task. Given a set of rules R and a set of literals S (plain literals and deontic literals), we can use the inference mechanism of defeasible logic (Section 2) to compute the set of conclusions (obligations) in force given the set of literals. These are the obligations an agent has to obey to in the situation described by the set of literals. However, the situation could already be sub-ideal, i.e., such that some of the obligations prescribed by the rules are already violated. Thus, given a set of literals describing a state-of-affairs one has to compute not only the current obligations, but also what reparation chains are in force given the set.

Consider a scenario where we have the rules $A \Rightarrow OB$ and $\neg B \Rightarrow OC$, and the effects are A and $\neg B$. The normal form of the rules is $A \Rightarrow OB \otimes OC$ and $\neg B \Rightarrow OC$. The only obligation in force for this scenario is OC. Since we have a violation of the first rule ($A \Rightarrow OB$ and $\neg B$), then we know that it is not possible to have an ideal situation here. Hence, computing only the current obligation does not tell us the state of the corresponding process. What we have to do is to identify the chain for the ideal situation for the task at hand. To deal with issue we have to identify the *active* reparation chains.

Some notational conventions. Given a rule r, $A(r)$ denotes the set of premises of the rules, and $C(r)$ the conclusion. For any set of rules R, $R[C]$ denotes the subset of R of rules whose conclusion is C. If $C = p_1 \otimes \cdots \otimes p_n \otimes q$ is a *reparation chain*, we use $\pi_i(C)$ to denote the i-th element of the chain.

Then, a reparation chain C is *active* given a set of literals S, if

1. $\exists r \in R[C] : \forall a_r \in A(r), a_r \in S$ and
2. $\forall s \in R[D]$ such that $\pi_1(C) \in D$, either
   1. $\exists a_s \in A(s) : \sim a_s \notin S$, or
   2. $\exists i\ \pi_i(D) = \sim \pi_1(C)$ and $\exists k$, $k < i$, $\sim \pi_k(D) \notin S$, or
   3. $\exists t \in R[E]$: $\pi_j(E) = \pi_1(C)$, $\forall a_t \in A(t), a_t \in S$, $\forall m$, $m < j$, $\sim \pi_m(E) \in S$ and $t \prec s$.

Let us examine the following example. Consider the rules

$$r_1 : A_1 \Rightarrow OB \otimes OC, \qquad r_2 : A_2 \Rightarrow O\neg B \otimes OD, \qquad r_3 : A_3 \Rightarrow OE \otimes O\neg B.$$

The situation S is described by A_1 and A_3. In this scenario the active chains are $OB \otimes OC$ and $OE \otimes O\neg B$. The chain $OB \otimes OC$ is active since A_1 is in S and r_2 cannot be used to activate the chain $O\neg B \otimes OD$. For r_3 and the resulting chain $OE \otimes O\neg B$, we do not have the violation of the primary obligation OE of the rule (i.e., $\neg E$ is not in S), so the obligation $O\neg B$ is not entailed by r_3.

4.2 Checking Compliance

A reparation chain is in force if there are a rule of which the reparation chain is the consequent and a set of facts (effects of a task in a process) including the rule antecedents. In addition we assume that, once in force, a reparation chain remains as such unless we can determine that it has been violated or the obligations corresponding to it have all been obeyed to (these are two cases when we can discharge an obligation or reparation chain). This means that it is not possible to have two instances at the same time of the same reparation chain. Accordingly, a reparation chain in force is uniquely determined by the combination of the task T when the chain has been derived and the rule R from which the chain has been obtained.

The procedure for compliance checking is based on two algorithms, *ComputeObligations* and *CheckCompliance*. *ComputeObligations* is the procedure given in the Section 4.1 to compute the set of active chains. Given a set of literals S, corresponding to effects of a task T in a process model, we use the algorithm *ComputeObligations* to determine the current set of active chains for the process *Current*. The set of the current active chains includes the new chains triggered by the task, as well as the chains carried out from previous tasks. The algorithm *CheckCompliance* scans all elements of *Current* against the set of literals S, and determines the state of each reparation chain ($C = A_1 \otimes A_2$) in *Current*. *CheckCompliance* operates as follows:

```
if $A_1 = OB$, then
  if $B \in S$, then
    remove($[T, R, A_1 \otimes A_2]$, $Current$)
    remove($[T, R, A_1 \otimes A_2]$, $Unfulfilled$)
    if $[T, R, B_1 \otimes B_2 \otimes A_1 \otimes A_2] \in Violated$ then
      add($[T, R, B_1 \otimes B_2 \otimes A_1 \otimes A_2]$, $Compensated$)
  if $\neg B \in S$, then
    add($[T, R, A_1 \otimes A_2]$, $Violated$)
    add($[T, R, A_2]$, $Current$)
  else
    add($[T, R, A_1 \otimes A_2]$, $Unfulfilled$).
```

Let us examine the *CheckCompliance* algorithm. Remember the algorithm scans all active reparation chains one by one, and then for each of them reports on the status of it. For each chain in *Current* (the set of all active chains), it looks for the first element of the chain and it determines the content of the obligation (so if the first element is OB, the content of the obligation in B). Then it checks whether the obligation has been fulfilled (B is in the set of effects), or violated ($\neg B$ is in the set of effects), or simply we cannot say anything about it (none of B and $\neg B$ is in the set of effects). In the first case we can discharge the obligation and we remove the chain from the set of active chains (similarly if the obligation was carried over from a previous task, i.e., it was in the set *Unfulfilled*). In case of a violation, we add the information about it in the system. This is done by inserting a tuple with the identifier of the chain and what violation we have in the set *Violated*. In addition, we know that violations can be compensated, thus if the chain has a second element we remove the violated element from the chain and put the rest of the chain in the set of active chains. Here we take the stance that a violation does not discharge an obligation, thus we do not remove the chain from the set of active chains[5]. Finally in the last case, the set of effects does not tell us if the obligation has been

[5] [16] propose a more fine grained classification of obligations, accordingly it is possible to have obligations that are discharged when are violated, as well as obligations that persist in case of

fulfilled or violated, so we propagate the obligation to the successive tasks by putting the chain in the set *Unfulfilled*. The algorithm also checks whether a chain/obligation was previously violated but it was then compensated.

The conditions below relate the state of a process based as reported by the *CheckCompliance* algorithm and the semantics for PCL expressions. In particular, a process is compliant if the situation at the end of the process is at least sub-ideal (it is possible to have violations but these have been compensated for). Similarly a process is fully compliant if it results in an ideal situation.

- A process is *compliant* iff for all $[T,R,A] \in \mathit{Current}$, $A = OB \otimes C$, for every $[T,R,A,B] \in \mathit{Violated}$, $[T,R,A,B] \in \mathit{Compensated}$ and $\mathit{Unfulfilled} = \emptyset$.
- A process is *fully compliant* iff for all $[T,R,A] \in \mathit{Current}$, $A = OB \otimes C$, $\mathit{Violated} = \emptyset$ and $\mathit{Unfulfilled} = \emptyset$.

Accordingly, a process is not compliant if the set of unfulfilled obligations (*Unfulfilled*) is not empty. Consider, for example the rule

$$r_3 : \mathit{account}(y) \Rightarrow \mathit{OpositiveBalance}(y) \otimes \mathit{OapproveManager}(y)$$

relative to the process of Figure 1 with the annotation as in Table 1. After task E we have, among others, the effect $\mathit{account}(y)$. This means that after task E we have the chain

$$[E, r_4, \mathit{OpositiveBalance}(y) \otimes \mathit{OapproveManager}(y)]$$

in *Current* for task F. After task F, the above entry for the chain obtained from rule r_4 is moved to the set *Unfulfilled*. Suppose now that tasks G and H do not have any annotation attached to them. In this case at the end of the process we still have the active chain, but the resulting situation is not ideal: the antecedent of the rule is a subset of the set of effects, but we do not have the first element of the chain as one of the effects. Thus, the process is not compliant.[6]

5 Related Work

This paper provides a means of investigate the impact of compliance controls on agents' process and of assisting in compliance checking, analysis and feedback for subsequent

a violation. The above algorithm can be easily modified to deal with the different types of obligations examined by [16].

[6] What about a situation where, after task F, we have a task producing the annotation $\mathit{approveManager}(x)$ but no task with effect $\mathit{positiveBalance}(x)$? Is the resulting process compliant? In this case we have the reparation of the violation, but not the violation. The issue here is that we could have that a sanction is enforced before the occurrence of the violation which the sanction was supposed to compensate. Thus we are in a situation similar to that described in footnote 5 where the way to address the issue depends on the types of the obligations we have to deal with. Anyway, (i) it is easy to modify algorithm *CheckCompliance* to account for this type of cases, (ii) if one accepts preemptive reparations one can change the definition that classifies a process as compliant by replacing the condition that $\mathit{Unfulfilled} = \emptyset$ with the condition: let S be the set of effects for the end task of a process, $\forall [T, R, OA_1 \otimes \cdots \otimes OA_n] \in \mathit{Unfulfilled}$, $\exists A_i \in S$.

(re)design of the processes. The procedure is based on efficient algorithms and is able to deal with reparation chains of deontic statements.

Research on compliance has carried out in the field of autonomous agents [2], but the majority of works are found in related areas, in particular on control modelling. [14] presents the logical language PENELOPE, that provides the ability to verify temporal constraints arising from compliance requirements on effected business processes. [22] develops a method to check compliance between object lifecycles that provide reference models for data artifacts e.g. insurance claims and business process models. [13] provides temporal rule patterns for regulatory policies, although the objective of this work is to facilitate event monitoring rather than the usage of the patterns for support of design time activities. Furthermore, [1] presented an architecture for supporting Sarbanes-Oxley Internal Controls, which include functions such as workflow modelling, active enforcement, workflow auditing, as well as anomaly detection.

Another line of investigation studies compliance based on the structure of business processes. [12] consider an approach where the tasks of a business process model, written in BPMN, are annotated with the effects of the tasks, and a technique to propagate and cumulate the effects from a task to a successive contiguous one is proposed. The technique is designed to take into account possible conflicts between the effects of tasks and to determine the degree of compliance of a BPMN specification. [8], on the other hand, investigate compliance in the context of agents and multi-agent systems based on a classification of paths of tasks. [25] proposed Concurrent Transaction Logic to model the states of a workflow and presented some algorithms to determine whether the workflow is compliant. [31] proposes a polynomial time algorithm to perform compliance checking of business processes. The algorithm propagates the effects of tasks from one task to the tasks following it. Norms are represented as logical clauses. The major limitation of these approaches to compliance is that they ignore the normative aspects of compliance.

There has been some complementary work in the field. analysis of formal models representing normative notions. [11] studies the performance of business contract based on their formal representation in event calculus. [10] seeks to provide support for assessing the correctness of business contracts represented formally through a set of commitments. The reasoning is based on value of various states of commitment as perceived by cooperative agents.

PCL has already been proposed to study compliance. [17] uses it to model business contracts and their compliance with BPMN process models based on the ideal semantics of [18]. In [19] we extend the work of [31] to check compliance against PCL representation of the norms a process has to comply with. The focus is on the propagation of effects and normative positions across tasks. The propagation algorithm is based on heuristic, and thus it gives an approximate compliance checking.

Also, there have been recently some efforts towards support for process modelling against compliance requirements. [32] provides a method for integrating risks in business processes. The proposed technique for “risk-aware” business process models is developed for EPCs (Event-Driven Process Chains) using an extended notation. [26] proposes an approach based on control tags to visualize internal controls on process models. [23] takes a similar approach of annotating and checking process models against compliance rules, although the visual rule language (BPSL) does not directly address the deontic notions providing compliance requirements.

Acknowledgements

This paper extends the work presented in [20].

NICTA is funded by the Australian Government as represented by the Department of Broadband, Communications and the Digital Economy and the Australian Research Council through the ICT Centre of Excellence program.

References

1. R. Agrawal, C. M. Johnson, J. Kiernan, and F. Leymann. Taming compliance with Sarbanes-Oxley internal controls using database technology. In *Proc. ICDE 2006*, 2006.
2. M. Alberti, M. Gavanelli, E. Lamma, F. Chesani, P. Mello, and P. Torroni. Compliance verification of agent interaction: a logic-based software tool. *Applied Artificial Intelligence*, 20(2-4):133–157, 2006.
3. G. Antoniou, D. Billington, G. Governatori, and M. J. Maher. Representation results for defeasible logic. *ACM Transactions on Computational Logic*, 2(2):255–287, 2001.
4. Grigoris Antoniou, David Billington, Guido Governatori, and Michael J. Maher. Embedding defeasible logic into logic programming. *Theory and Practice of Logic Programming*, 6(6):703–735, 2006.
5. W. Binder and et al. A multiagent system for the reliable execution of automatically composed ad-hoc processes. *JAAMAS*, 2006.
6. G. Boella and L. van der Torre. Fulfilling or violating obligations in multiagent systems. In *Procs. IAT04*, 2004.
7. J. Carmo and A.J.I. Jones. Deontic logic and contrary to duties. In D.M. Gabbay and F. Guenther, editors, *Handbook of Philosophical Logic, 2nd Edition*. Kluwer, 2002.
8. Amit K. Chopra and Munindar P. Sing. Producing compliant interactions: Conformance, coverage and interoperability. In *Declarative Agent Languages and Technologies IV*, pages 1–15, 2007.
9. F. de Jonge, N. Roos, and C. Witteeven. Primary and secondary diagnosis of multi-agent plan execution. *JAAMAS*, 2008.
10. N. Desai, N. C. Narendra, and M. P. Singh. Checking correctness of business contracts via commitments. In *Proc. AAMAS 2008*, 2008.
11. A. D. H. Farrell, M. J. Sergot, M. Sallé, and C. Bartolini. Using the event calculus for tracking the normative state of contracts. *International Journal of Cooperative Information Systems*, 14, 2005.
12. Aditya Ghose and George Koliadis. Auditing business process compliance. In *Service Oriented Computing, ISOC 2007*, LNCS, pages 169–180. Springer, 2007.
13. C. Giblin, S. Müller, and B. Pfitzmann. From regulatory policies to event monitoring rules: Towards model driven compliance automation. Technical report, IBM Zurich Research Lab., 2006.
14. S. Goedertier and J. Vanthienen. Designing compliant business processes with obligations and permissions. In *Business Process Management (BPM) Workshops*, 2006.
15. G. Governatori. Representing business contracts in RuleML. *International Journal of Cooperative Information Systems*, 14(2-3):181–216, 2005.

16. G. Governatori, J. Hulstijn, R. Riveret, and A. Rotolo. Characterising deadlines in temporal modal defeasible logic. In *Proc. Australian AI 2007*, 2007.
17. G. Governatori, Z. Milosevic, and S. Sadiq. Compliance checking between business processes and business contracts. In *Proc. EDOC 2006*, 2006.
18. G. Governatori and A. Rotolo. Logic of violations: A Gentzen system for reasoning with contrary-to-duty obligations. *Australasian Journal of Logic*, 4:193–215, 2006.
19. Guido Governatori, Jörg Hoffmann, Shazia Sadiq, and Ingo Weber. Detecting regulatory compliance for business process models through semantic annotations. In *4th International Workshop on Business Process Design*.
20. Guido Governatori and Antonino Rotolo. An algorithm for business process compliance. In Enrico Francesconi, Giovani Sartor, and Daniela Tiscornia, editors, *Legal Knowledge and Information Systems*, pages 186–191. IOS Press, 2008.
21. Guido Governatori and Shazia Sadiq. The journey to business process compliance. In Jorge Cardoso and Wil van der Aalst, editors, *Handbook of Research on BPM*, chapter 20, pages 429–457. IGI Global, 2009.
22. J. M. Küster, K. Ryndina, and H. Gall. Generation of business process models for object life cycle compliance. In *Proc. BPM 2007*, 2007.
23. Y. Liu, S. Müller, and K. Xu. A static compliance-checking framework for business process models. *IBM Systems Journal*, 46(2):335–362, 2007.
24. Donald Nute. Defeasible logic. In *Handbook of Logic in Artificial Intelligence and Logic Programming*. 1994.
25. Dumitru Roman and Michael Kifer. Reasoning about the behaviour of semantic web services with concurrent transaction logic. In *VLDB*, pages 627–638, 2007.
26. S. Sadiq, G. Governatori, and K. Naimiri. Modelling of control objectives for business process compliance. In *Proc. BPM 2007*, 2007.
27. Shazia Sadiq and Guido Governatori. A methodological framework for aligning business processes and regulatory compliance. In Jan van Brocke and Michael Rosemann, editors, *Handbook of Business Process Management*. Springer, 2009.
28. P. Stone and M. Veloso. Task decomposition, dynamic role assignment, and low-bandwidth communication for real-time strategic teamwork. *Artificial Intelligence*, 1999.
29. L. van der Torre, G. Boella, and H. Verhagen, editors. *Normative Multi-agent Systems*, Special Issue of *JAAMAS*, vol. 17(1), 2008.
30. J. Vanhatalo, H. Völzer, and F. Leymann. Faster and More Focused Control-Flow Analysis for Business Process Models through SESE Decomposition. In *Proc. ICSOC 2007*, 2007.
31. Ingo Weber, Guido Governatori, and Jörg Hoffmann. Approximate compliance checking for annotated process models. In Marta Indulska, Shazia Sadiq, and Michael zur Muehlen, editors, *Proceedings of the 1st International Workshop on Governance, Risk and Compliance — Applications in Information Systems (GRCIS'08)*, volume 339, pages 46–60. CEUR Workshop Proceedings, 17 June 2008.
32. M. zur Muehlen and M. Rosemann. Integrating risks in business process models. In *Proc. ACIS 2005*, 2005.

Massively multiple online role playing games as normative multiagent systems

Magnus Johansson and Harko Verhagen

K2lab, Dept. of Computer and Systems Sciences,
Stockholm University/KTH, Forum 100, 16440 Kista, Sweden
{magnus, verhagen}@dsv.su.se

Abstract. The latest advancements in computer games offer a domain of human and artificial agent behaviour well suited for analysis and development based on normative multi agent systems research. One of the most influential gaming trends today, Massively Multi Online Role Playing Games (MMORPG), poses new questions about the interaction between the players in the game. If we model the players and groups of players in these games as multiagent systems with the possibility to create norms and sanction norm violations we have to create a way to describe the different kind of norms that may appear in these situations. Certain situations in MMORPG are subject to discussions about how norms are created and propagated in a group, one such example involves the sleeper in the game Everquest, from Sony Online Entertainment (SOE). The Sleeper was at first designed to be unkillable, but after some events and some considerations from SOE the sleeper was finally killed. The most interesting aspect of the story about the sleeper is how we can interpret the norms being created in this example. We propose a framework to analyse the norms involved in the interaction between players and groups in MMORPG. We argue that our model adds complexity where we find earlier norm typologies lacking some descriptive power of this phenomenon, and we can even describe and understand the confusing event with the sleeper in Everquest.

Introduction

The games of today, both computer games and console games, are starting to focus on the opportunities that online co-operation can provide for the gaming experience. Games such as World of Warcraft (WoW) can have as many as thousands of active players in one of their gaming servers at the same time. Much of the “Massively multiple online role playing games” (MMORPG) genre seems to be all about co-operation and playing together and this in turn makes MMORPG:s an interesting phenomenon to investigate. In WoW there are many opportunities to engage in different social formations of different sizes, but one of the most common is to join a guild. A guild is a group of players that decide to play together for a period of time exceeding the length of one playing session. It is also possible to form smaller groups with short term goals.

Dagstuhl Seminar Proceedings 09121
Normative Multi-Agent Systems
http://drops.dagstuhl.de/opus/volltexte/2009/1895

After exploring the game world of WoW it is obvious that these games have rules, codes of conduct, do:s and don't:s that are either explicit or implicit. We may even want to call them norms, and these norms seem to be part of the very fabric of the interaction in this game genre. It is important to get an understanding for the differences between where a designer actually could influence the norms and where the norms are beyond the control of the designer and perhaps constantly evolving. If we take a close look at different aspects of most MMORPG:s it will be apparent that some parts of the game will live a life of its own, where local norms will appear through the interaction between players.

In "Ten Challenges for Normative Multiagent Systems" (Boella et al, 2008b), one of the examples comes from game playing, describing a team effort in Everquest to kill "The Sleeper", which was initially designed to be unkillable. This example is also used as the running example in (Boella et al. 2008a). Being interested in the design of games and following discussions online, it seems as though the killer actually was vulnerable to the attacks of the group due to a miss when updating one of the zones in Everquest, therefore a series of events led the sleeper to be killed by the group after some consideration from Sony, since one group came really close when the sleeper suddenly disappeared (this is thought to be the result of a Sony employee, resetting the instance when the impossible was about to happen). All other scenarios than this would indicate that the game designer would have made a big mistake in the initial efforts of designing the Sleeper. For more details we include the email send by Sony[1] and an explanation offered on Wikipedia[2] both focussing on a software glitch as the cause for the first Sleeper killing.

[1] "The Sleeper (11-17-03)

Over the weekend several guilds gathered on Rallos to fight with the Sleeper. Unfortunately, their encounter was cut short due to an apparent bug. I wanted to take a moment here and apologize to those that were there, and to those that have heard about the event through their friends. The bug concerned an NPC in the zone that appeared to have been causing the Sleeper to not focus on the player characters. The decision was made at the time to end the event. Further investigation has only served to make it unclear if this was a real issue or not.

I, on behalf of the company, apologize for any consternation this may have caused during your play time. If anyone is going to defeat the Sleeper, it should be done without any question about the validity of the event. We're very sorry that this first attempt was halted, but at the time it seemed like the best thing to do.

We have resurrected and restored those that participated. We have corrected the potential problem, and have reset the encounter. Other than that one potentially problematic NPC, nothing else about the encounter has been changed.

We want to wish those on Rallos that are planning to tackle the Sleeper the best of luck.

Send me some screenshots of all of you standing around the corpse, I'd love to post them on the site.

Thank you and thanks for understanding,

Alan" (EverQuest Chat)

[2] "Kerafyrm, "The Sleeper", is a dragon boss in the original "The Sleeper's Tomb" zone.

In this article we will introduce the view on norms as it has developed in the social sciences, mainly sociology. Then we will propose an extension to the normative framework developed by Gibbs (1965) and apply this framework to situations in WoW. Finally we will describe some related research before we finish with conclusions and proposals for future research.

Normative multiagent systems and the definition of norms in the social sciences

At the 2007 NorMAS Dagstuhl workshop the definition of a normative multiagent system was the concluding part of the whole week, together with a list of future research questions. In the introduction to the following JAAMAS special issue (Boella et al. 2008b) the definition voted for by the majority of participants is presented as:

> “A normative multiagent system is a multiagent system organised by means of mechanisms to represent, communicate, distribute, detect, create, modify, and enforce norms, and mechanisms to deliberate about norms and detect norm violation and fulfilment.”

Note that the definition does not define the nature of the agents (i.e., they can be artificial or human) nor about the boundaries of a normative system (even if it gives the impression of a well-defined system).

Within the social sciences and more particularly in sociology and social philosophy norms are discussed and defined in different ways. We present

While sleeping, Kerafyrm is guarded by four ancient dragons (warders) in "The Sleeper's Tomb". When all four dragons are defeated by players and are dead at the same time, The Sleeper awakes, triggering a rampage of death. Kerafyrm travels through and into multiple zones from The Sleeper's Tomb to Skyshrine, killing every player and NPC in his path. This event is unique in EverQuest, because it can only occur once on each game server. Once The Sleeper awakes, neither he nor the original guardians will ever appear again on that server, unless the event is reset by SOE.

As of 12 July 2008, Kerafyrm remains asleep only on the Al'Kabor (Macintosh) server.

Originally intended to be unkillable, SOE prevented a raid of several guilds on Rallos Zek server from potentially killing him because of a potential bug. SOE later apologized for interfering,[25] and allowed the players to retry the encounter.

"Kerafyrm The Awakened" appears in the expansion Secrets of Faydwer as part of a raid event "Crystallos, Lair of the Awakened" in the instanced zone of "Crystallos." ” (Wikipedia) some of the definitions common on the social sciences and conclude with the framework we will use.

In Gibbs (1965) a typology of norms concerning the regulation of behaviour and acts is described encompassing conventions, morals, mores, rules and laws as depicted in table 1. These various social mechanisms are structured using the following dichotomies:

- Probability that a sanction will be issued (yes – no)
- Characteristics of the agent issuing a sanction (special status or no special status)
- Evaluation of an act (collective or not)
- Expectation concerning the act (collective or not)

		Low probability of a possible sanction when the act occurs	High probability of a possible sanction when the act occurs			
			By anyone (i.e., without regard to status)		Only by a person or persons in a particular status or statuses	
evaluation of the act	*expectation concerning the act*		By means that exclude the use of force	By means that may include the use of force	By means that exclude the use of force	By means that may include the use of force
Collective evaluation	Collective expectation	Type A: Collective conventions	Type D: Collective morals	Type H: Collective mores	Type L: Collective rules	Type P: Collective laws
	No collective expectation	Type B. Problematic conventions	Type E: Problematic morals	Type I: Problematic mores	Type M: Problematic rules	Type Q: Problematic laws
No collective evaluation	Collective expectation	Type C: Customs	Type F: empty class	Type J: empty class	Type N: Exogenous rules	Type R: Exogenous laws
	No collective expectation	Logical null class, i.e., non-normative	Type G: empty class	Type K: empty class	Type O: Coercive rules	Type S: Coercive laws

Table 1. Gibbs' Norm typology (1965)

Tuomela (1995) on his turn distinguished two kinds of social norms (meaning community norms), namely, rules (r-norms) and proper social norms (s-norms). Rules are norms created by an authority structure and always based on agreement making. Proper social norms are based on mutual belief. Rules can be formal, in which case they are connected to formal sanctions, or informal, where the sanctions are also informal.

Proper social norms consist of conventions, which apply to a large group such as a whole society or socioeconomic class, and group-specific norms. The sanctions connected to both types of proper social norms are social sanctions and may include punishment by others and expelling from the group. Aside from these norms, Tuomela also described personal norms and potential social norms[3] containing, among others, moral and prudential norms (m-norms and p-norms, respectively). The reasons for accepting norms differ as to the kind of norms:

- Rules are obeyed because they are agreed upon.
- Proper social norms are obeyed because others expect one to obey.
- Moral norms are obeyed because of one’s conscience.
- Prudential norms are obeyed because it is the rational thing to do.

The motivational power of all types of norms depends on the norm being a subject’s reason for action. In other words, norms need to be internalized and accepted.

Therborn (2002) distinguishes three kinds of norms. Constitutive norms define a system of action and an agent's membership in it; regulative norms describe the expected contributions to the social system, and distributive norms defining how rewards, costs, and risks are allocated within a social system. Furthermore, he distinguishes between non-institutionalized normative order, made up by personal and moral norms in day-to-day social traffic, and institutions, an example of a social system defined as a closed system of norms. Institutional normative action is equalled with role plays, i.e., roles find their expressions in expectations, obligations, and rights vis-à-vis the role holder's behaviour.

In Elster (2007) a whole range of social mechanisms are described. Among them is the concept of social norms. A social norm is defined as an injunction to act or abstain from acting. The working mechanism is the use of informal sanctions aimed at norm violators. Sanctions may affect the material situation of the violator via direct punishment or social ostracism. An open question is the costs of sanctioning. Apart from social norms Elster describes moral norms (that are unconditional) and quasi-moral norms (like social norms these are conditional but triggered by being able to observe what others are doing instead of by being observed by other people as is the case for social norms). Other connected concepts are legal norms (where special agents enforce the norms) and conventions that are independent of external agent action. In his text, Elster discusses in detail some examples of norms such as: norms about etiquette, norms as codes of honour, and norms about the use of money.

[3] Potential social norms are norms that are normally widely obeyed but not in their essence based on social responsiveness and that, in principle, could be personal only.

Combining these frameworks results in the following: Therborns regulative norms encompass all of Gibbs categories whereas Therborns constitutive and distributive norms are outside of Gibbs' scope. Tuomela's r-norms encompass Gibbs type L, M, P and Q and his s-norms type D, E, H and I respectively. The moral norms Tuomela mentions are outside of Gibbs scope as these are norms where an agent is its own evaluator. Prudential norms are incommensurable with Gibbs typology or indeed any other typology. Elsters moral norms are equivalent to Tuomela's moral norms whereas his quasi-moral norms seem to fit to Gibbs type O and S. Elster's conventions map to Gibbs type A and the legal norms to type P or Q. This is presented in table 2 below.

In the remainder of this paper we will use the following notion of norms:

> "Norms are statements about the appropriateness of an agent's act which may result in a sanction being issued by another agent or an agent belonging to a specific class of agents."

		Low probability of a possible sanction when the act occurs	High probability of a possible sanction when the act occurs			
			By anyone (i.e., without regard to status)		Only by a person or persons in a particular status or statuses	
evaluation of the act	*expectation concerning the act*		By means that exclude the use of force	By means that may include the use of force	By means that exclude the use of force	By means that may include the use of force
Collective evaluation	Collective expectation	Elster conventions	Tuomela s-norms	Tuomela s-norms	Tuomela r-norms	Tuomela r-norms/ Elster legal norms
	No collective expectation	Type B. Problematic conventions	Tuomela s-norms	Tuomela s-norms	Tuomela r-norms	Tuomela r-norms/ Elster legal norms
No collective evaluation	Collective expectation	Type C: Customs	Type F: empty class	Type J: empty class	Type N: Exogenous rules	Type R: Exogenous laws
	No collective expectation	Logical null class, i.e., non-normative	Type G: empty class	Type K: empty class	Elster quasi-moral norms	Elster quasi-moral norms

Table 2. Adapted version of Gibbs’ norm typology (equivalent to Therborns regulative norms) encompassing Elsters and Tuomelas norm typologies.

Norms in MMORPG

We propose to use the revised framework presented above to understand the dynamics of the most common norms and norm violations in MMORPG:s.

In MMORPG:s severe violations are usually punished by ostracisation of the norm violators or the loss of points in a value system where a player can earn points for assisting the guild in raids (measured in DKP, short for Dragon Killing Points). It may be difficult to differentiate between what social behaviour is acceptable and what is not.

Some players exhibit behaviour that violates norms in ways that could be described as cheating or grief play. Some of these examples are so common that most guilds have structured their rules to cover these issues as well. Smith (2004) mentions three different categories of behaviours that might infringe on the gaming experience of others. The three categories are cheating, local norm violation and grief play.

Cheating

“Cheating” is the use of any technique that runs against the spirit of the game without being technically unfair (in the sense that it is violating a norm). It is however difficult to prove whether or not someone is cheating. The risk of sanctions being made against a violator depends on the severity of the violation. If the violation is very severe there usually is a “High probability that an attempt will be made to apply a sanction when the act occurs” (from Gibbs typology (1965) corresponding to Tuomela s-norms).

Local norm violation

“Local norm violation” is any violation of a mutual understanding of how the game ought to be played. These actions have different level of implications for other players and the players are usually sanctioned if the violation appears repeatedly. These violations have a “High probability that an attempt will be made to apply a sanction when the act occurs” from Gibbs typology (1965) but we have to keep in mind that minor violations might be ignored. These actions could potentially be sanctioned by anyone in the group, but the most probable solution in the case of a raid group would be that the raid leader would solve the problem without the use of force. The severest forms of violations may be punished with ostracism.

Grief play

"Grief play" is a broad category of behaviour which causes a severe and stressful disadvantage to the target. Examples of grief play are; unprovoked harassment through game chat channels, repeatedly killing a player as soon as the character comes back to life, and behaviour not related to the winning condition of the game. Grief play in its different forms is behaviour that infringes the higher level norms of the realm and can be difficult to sanction. The penalty for someone engaging in this kind of activity should perhaps be ostracisation, but since the players are from different factions, it is difficult to make any sanctions from the victim's side. Grief play would therefore fit the description of "Low probability that an attempt will be made to apply a sanction when the act occurs" from Gibbs typology (1965).

All examples above are examples of social norms, since norm violations are punished with sanctions and are thus in accordance with e.g., Elster and our definition of norms. In the case of the last example this can be hard to prove however. The typology taken from Gibbs gives a better understanding at least when it comes to the probability of a sanction to occur, but it is very difficult to judge from case to case, since all these violations have different severity and impact on other players. Thus it seems that Gibbs framework and consequently also our revised framework may need to be extended to produce a more fine grained categorisation.

Norms regulating the use of money

Not surprisingly, money and valuable equipment may lead to conflicts in MMORPG:s. There are multiple ways of breaching norms for how to distribute money and equipment between all members of a guild. Some of the most common examples where discussions about money occur are the following situations; begging, ninja looting, and twinking.

Begging

Begging is usually other gamers in game asking for money, and this can in fact be disturbing behaviour that many guilds have strict rules against. Most beggars are being ostracised or ignored, since it is hard to make other sanctions against them. Beggars will eventually earn a bad reputation since gamers will gossip about this unwanted behaviour. It may be argued that this is addressed by Thernborns distributive norms.

Ninja looting

Ninja looting is another form of misconduct that most guilds have rules against. When a gamer steals the loot from another gamer under certain conditions when playing as a group this is defined as ninja looting.

Both begging and ninja looting have a “High probability that an attempt will be made to apply a sanction when the act occurs”, ostracism is the most probable action taken, but other actions may occur. The probability of sanctions including force is not very probable. Ninja looting can also be seen as a breach against Therborns distributive norms thus placing it outside the set of regulative norms.

Twinking

Twinking is when a high level gamer decides to help a low level character with money to buy better equipment or helping the low level gamer killing creatures above his/her skill level.

The last example is actually not a serious norm violation and most gamers do not care about it and thus it would fit in the first category of Gibbs’ typology (1965) where no sanction would appear. It would also fit in under Therborns distributive norms.

Norms regulating the use of tools

Most MMORPG:s today are highly complex and sometimes a player can find that it is hard to keep track of the situation in game. Most games with a certain degree of complexity will eventually be subject to “add-ons”, where someone develops tools to highlight information in the game or perhaps give certain advantages for a player with the add-on installed.

Add-ons range from small “cheating” applications in games such as “Counter strike” where “auto aiming” and the possibility to see through walls were used by some players. In WoW the most common add-ons are used for co-ordinating raid groups and displaying statistics for all characters in an instance (both players and Mob). This gives all players in the group an advantage that is not considered unfair, since most players use this kind of tools. But what is interesting is where to draw the line of what is considered enhancing the game and what is considered cheating. Norms are usually subject to constant change and there are interesting stories where new forms of norms are being created.

T L Taylor (2006) describes the use of a tool called CTRaidAssist during a raid. This tool monitors many statistics of the characters of a raid group and in this example someone in the group came a bit to close to a mob (a non-player character or NPC) and therefore the entire group was being attacked by the mob and nearly killed. The raid leader (using CTRaidAssist) could see that the amount of aggression (a measurement of how close or threatening a character is to a mob) had increased, which had triggered the attack. The interesting part about this story is that the raid leader told everyone in the raid group that if someone would do the same thing again, this would result in penalties. This shows that tools can be used to monitor the players’ behaviour and thus enable the possibility of sanction behaviour that previously could not be sanctioned. This involves a move from one category in Gibbs framework to another. Without the tool there is no or a very low probability of a sanction since the action cannot be detected. The tool enables a special person (in this case the raid leader) to issue a sanction. The message send by the leader leads to a collective expectation that players will refrain from this action and it is only the leader that can evaluate so there is no collective evaluation. So, introducing the tool moves the raid group from the logical null-class (non-normative situation) in Gibbs' typology to the situation labelled as "exogenous rules" (type N) even if the rules and sanctioning agent are mutually agreed upon in and part of the group.

Different levels of organisation where norms appear in MMORPG:S

WoW can be described at different organisational levels and as different types of norm systems, ranging from a high level perspective (the different types of servers, usually called realms in the game) down to the lowest level focusing on players and small groups. What seem to be characteristic about the higher levels such as the different gaming realms and factions is that the norms are of a wider scope, and communicate the spirit of the game without much attention to detail. On the middle level (Guilds) there seems to be a stricter way of communicating, creating, and changing norms. It is apparent that a large group needs some form of organisation to work properly. On the lowest level (groups) there seems to be a mutual respect for the group and the norms are close to what could be considered common sense. The difference between the highest level and all levels below is that sanctions are more easily distributed on the lower levels, perhaps because they are agreed upon within a group with a finite number of players in a way similar to the proper social norms discussed by Tuomela (1995).

Game servers

The different types of game servers give rise to different sets of norms for the type of interaction that takes place on the server. Three different kinds of servers will be mentioned here, since they are the most common:

1. Normal servers (No special rules applied),
2. PvP servers (Player versus Player), and
3. RP servers (Role Playing Servers).

There are combinations of these types of servers, but they will not be discussed here since these combinations do not interfere with our analysis of the basic types.

For our purpose the most interesting types of servers are the fairly restricted RP servers where all players are to stay in character when playing. This means that the player has to play along and make decisions according to what would be most likely for the character in the game. For instance, discussing game functionality or other meta-gaming issues is not allowed on these servers, since it would interfere with the overall gaming experience.

Normal servers are servers where no explicit rules are applied. This gives players a freedom from the strict rules of the RP servers which could possibly lead to a different kind of interaction. The special rules on the level of game servers are an example of the constitutive norms as described by Therborn.

Factions

All MMORPG:s have some kind of history and a world with resources that are being shared between its inhabitants in one way or the other. For the sake of making this history interesting a player belong to a faction. In WoW one is associated with either the Horde faction or the Alliance faction depending upon the race choosen during the character creation process. On all types of servers it would be fair to kill a character from the opposing faction. But there are specific norms on what is acceptable and what is not. For instance, a high level player who kills someone from the opposing faction who does not stand a chance of defending him/herself would be regarded as playing unfair, or even as a performing grief play, and may, if repeated, lead to a stressful disadvantage for the target.

Groups (Guilds and small groups)

Groups in WoW may lead to observable behaviour and sometimes conflicts. Guilds usually have a forum page where all issues concerning in game tactics are being discussed. Rules are usually available in the forum pages of guilds, to inform all players of the norms that all players should stick to.

Large groups/Guilds

Guilds are large group of players that play together often aiming at co-operating in so called raid groups. A raid group consists of as many as 25 players co-operating to overcome Non player characters (NPC) in special instances of the game.

Small groups

Small groups can consist of 2 or more players co-operating on small missions in game, called quests. In WoW, it’s sometimes apparent that the quests are too hard for a single player and that joining a group is the only solution to solve the quest.

		Low probability of a possible sanction when the act occurs	High probability of a possible sanction when the act occurs			
			By anyone (i.e., without regard to status)		Only by a person or persons in a particular status or statuses	
evaluation of the act	*expectation concerning the act*		By means that exclude the use of force	By means that may include the use of force	By means that exclude the use of force	By means that may include the use of force
Collective evaluation	Collective expectation	Elster conventions	Tuomela s-norms **interaction in small groups and in guilds, cheating, and** **local norm violation**	Tuomela s-norms **interaction in small groups and in guilds, cheating, and** **local norm violation**	Tuomela r-norms	Tuomela r-norms/ Elster legal norms
	No collective expectation	Type B. Problematic conventions **Grief play**	Tuomela s-norms **cheating, and** **local norm violation**	Tuomela s-norms **cheating, and** **local norm violation**	Tuomela r-norms	Tuomela r-norms/ Elster legal norms
No collective evaluation	Collective expectation	Type C: Customs	Type F: empty class	Type J: empty class	Type N: Exogenous rules **Guildleader using CTRaid Assist**	Type R: Exogenous laws
	No collective expectation	Logical null class, i.e., non-normative	Type G: empty class	Type K: empty class	Elster quasi-moral norms	Elster quasi-moral norms

Table 3. Categorization of the examples gien our adapted version of Gibbs' norm typology.

Related research

In Boella et al. (2008a) the EverQuest example described above is used as a running example in an analysis of norm negotiation in online multi-player games. The authors describe a two step negotiation process where the first step consists of negotiating a social goal and the second step is negotiating the norms and sanctions. Thus the setting is an argumentative one. However the norms are statements over goals rather than acts. Since this gives the agent freedom on how to obtain a goal there is no expectation concerning acts, only with regard to obtaining a goal. The authors also accept norms not connected to sanctions. If we put this research into Gibbs typology of norms then it would fit into the category “Logical null: i.e., non-normative.” Since we base our research on Gibbs and other social science theories on norms, the definition of norms (Boella et al. 2008a) use is not compliant with our definition because we include only norms connected to possible sanctions and as evaluations of acts rather than goals. A problem with goals is of course that one never knows if a goal will be obtained thus without the concept of time sanctioning is impossible. If norms instead are seen as addressing acts as they occur the evaluation is independent of any projections into the future. The tools proposed in (Boella et al. 2008a) address the issue of communication between agents at the level of goals.

Conclusions and discussion

We have introduced the reader to an extended version of the norm categorisation scheme developed by Gibbs (1965). In our examples we have shown that this framework enhances our understanding of human MMORPG gamer behaviour and that we can include the norms of agents external to the normative multiagent systems in the framework. Whatever the true explanation for the Sleeper example may be, be it an extreme form of collaboration or a consequence of a software glitch, it illustrates the point of the importance of the norms imposed from outside the agent system, namely the norms of the designers. If the software glitch explanation is true we may see this as move from the situation in L/P in Gibbs typology (i.e., only persons with a particular status may sanction the outcome of an action given the collective evaluation of the act and the collective expectation towards the act, also presuming Sony and the players to form one shared system) to either situations O/S (i.e., only persons with a particular status may sanction the outcome of an action given there is no collective evaluation of the act and no collective expectation towards the act, thus presuming Sony and the players do not form one shared system) or even to the logical null class (same as previous categorisation only with a low probability there will be a sanction).

The framework itself suggests that tools for normative multiagent systems should include possibilities to monitor behaviour, moving the whole system to another part of the categorisation matrix by enabling sanctioning. We propose that the extended framework needs to be developed further to a finer grained categorisation to deal with the (close to) real world phenomena encountered in MMORPG.

References

Boella, G., Caire, P., and van der Torre, L.: Norm negotiation in online multi-player games, Knowledge Information Systems,, online at http://dx.doi.org/10.1007/s10115-008-0162-2, (2008a)

Boella, G., van der Torre, L., and Verhagen, H.: Introduction to the special issue on normative multiagent systems, JAAMAS 17 (1), pp. 1 – 10, (2008b**)**

Elster, J.: Explaining Social Behavior – More Nuts and Bolts for the Social Sciences. Cambridge University Press, (2007)

EverQuest Chat, Nov 24 2003, Nov 24th 2003, http://web.archive.org/web/20031121164036/eqlive.station.sony.com/community/dev_view.jsp?id=59485, last checked January 20, 2009.

Gibbs, J.P.: Norms: The Problem of Definition and Classification. The American Journal of Sociology 70, 586 – 594, (1965)

Smith, J.H.: Playing dirty - understanding conflicts in multiplayer games. Paper presented at the 5th Annual Conference of the Association of Internet Researchers (AoIR), Brighton, UK, 19—22 September. http://www.itu.dk/people/smith/texts/playing_dirty.pdf, last checked January 20, 2009.

Taylor, T.L.: Does WoW change everything?, Games and Culture, 1 (4), (2006)

Therborn, G.: Back to Norms! On the Scope and Dynamics of Norms and Normative Action. Current Sociology 50, 863 – 880, (2002)

Tuomela, R.: The Importance of Us: A Philosophical Study of Basic Social Norms. Stanford University Press, (1995)

Wikipedia, EverQuest, http://en.wikipedia.org/wiki/Everquest, last checked January 20, 2009

Modeling and Validating Norms[*]

Viviane Torres da Silva Christiano Braga

Computer Science Department, Universidade Federal Fluminense (UFF)
Rua Passos da Pátria 156, Bloco E, 24210-240, Niterói, Brazil
{viviane.silva, cbraga}@ic.uff.br

Abstract. Norms describe the permissions, prohibitions and obligations of agents in multi-agent systems in order to regulate their behavior. In this paper we propose a normative modeling language that makes possible the modeling of norms motivating the modeling of such norms together with the non-normative part of the system. In addition, we also propose a mechanism to validate the norms at design time, i.e., to check if the norms respect the constraints defined by the language and also their possible conflicts.

Keywords: norm, modeling, validation, conflict, metamodel.

1 Introduction

Norms are used to regulate the behavior of the agents in open multi-agent systems (MAS) by describing their permissions, prohibitions and obligations. The definition of norms is an important part of the specification of a system and should be treated as an important task of MAS design. Methodologies such as Gaia [29][29], MaSE [5], SODA [20] and PASSI [3] [10] propose the specification of organization rules (or norms) during the analysis phase and recognize the need to associate these rules with design elements. However, there are still few modeling languages that support the modeling of norms together with the modeling of the entities that compose a MAS.

It is important to consider norms while designing a MAS since:

(i) *Norms refer to actions, agents and roles that compose a system.* They specify the actions that agents playing roles in the system are obliged, permitted or prohibited to execute. Therefore, redesigning the system, for instance, by excluding a role, may affect the norms. On the other hand, the definition of a new norm will only be possible if the actions, agents and roles being mentioned in the norm are being considered in the system design.

(ii) *Norms' conflicts can cause the redesign of a system.* Two norms are in conflict, for instance, if one gives a permission and another a prohibition to an agent to execute the same action in the same time frame. When it occurs, it is necessary to rewrite one of the norms in order to eliminate the conflict. While rewriting the norm, it may be desired, or even necessary, to redesign the system.

[*] The present work has been partially funded by the Spanish project "Agreement Technologies" (CONSOLIDER CSD2007-0022) and by the Brazilian research council CNPq under project no 550865/2007-1.

Dagstuhl Seminar Proceedings 09121
Normative Multi-Agent Systems
http://drops.dagstuhl.de/opus/volltexte/2009/1918

The two main goals of this paper are: (i) to support norm modeling during the design phase of a MAS and (ii) to define a technique to check possible conflicts between two defined norms at design time. We propose a normative modeling language, called NormML, that can be used during the design phase of a MAS to model the corresponding norms and an invariant-based technique to check for *well-formedness* of the norms and conflicts between two norms. Such invariants are defined over the *metamodel* of NormML.

The novelty of our approach is twofold: first, the modeling language itself, to model norms and second a validation technique that, when supported by a tool (Section 3.3), can automatically check conflicts between norms at design-time. None of the proposed methodologies or modeling languages for MAS is able to represent the three norm kinds (permission, obligation and prohibition) and to check their conflicts.

This paper is organized as follows. Section 2 provides some background material and Section 3 introduces our normative modeling language and tool used to automatically check for conflicts and query the norms model. In Section 4 we present related work. Section 5 concludes the paper with final remarks and discusses future work.

2 Background

NormML is a modeling language to specify norms that constraint the behavior of agents in MAS. Our modeling language was designed with the perception that *norm specification in MAS design and security policy specification in role-based access control (RBAC)* [10] *design are closely coupled issues*. RBAC security policies specify the *permissions* that a *user* has under a given *role*, while trying to access system *resources*. In MAS we specify the *norms* that regulate the *behavior* (or actions) of an *agent* playing a given *role*.

In this section we briefly provide background material for the rest of this paper. In Section 2.1 we introduce the necessary norm-related terminology that will be used throughout the paper. Section 2.2 introduces basic notions of models and metamodels, necessary to understand the design of NormML. In Section 2.3 we introduce Secure UML [1], a UML-based [18] modeling language for RBAC, which we extend with normative-related concepts. Such an extension gives rise to NormML.

2.1 Norms

A norm can be used to regulate the interaction between two agents—those norms are called dialogical norms [10]—and to regulate the access to resources, the entering and leaving of agents in organizations and environments, and the permissions to play roles.

A norm describes an action that is being permitted, obligated or prohibited, the entity whose behavior is being regulated (an agent, a role or an agent playing a given role) and a set of conditions to activate and deactivate the norm.

2.2 Models and metamodels

A modeling language provides a vocabulary (concepts and relations) for creating *models*. Such vocabulary is described by the *metamodel* of the modeling language which elements formalize the language concepts and their relationships. A metamodel may include invariants that specify additional properties that the models must fulfill as instances of the metamodel. Such invariants may specify the *well-formedness* conditions of a model with respect to its metamodel and the *consistency* conditions between metamodel concepts.

When UML is chosen as metalanguage, a metamodel is represented by a class diagram and its invariants are written in OCL (Object Constraint Language) [17]. This is the choice followed in this paper.

2.3 Secure UML

Secure UML provides a language for modelling *Roles*, *Permissions*, *Actions*, *Resources*, and *Authorization Constraints*, along with the relationships between permissions and roles, actions and permissions, resources and actions, and constraints and permissions. The actions described in the language can be either *Atomic* or *Composite*. The atomic actions are intended to map directly onto actual operations of the modeled system (delete, update, read, create and execute). The composite actions are used to hierarchically group atomic ones.

SecureUML leaves open what the protected resources are and which actions they offer to clients. ComponentUML [1] is a simple language for modeling component-based systems that provides provides a subset of UML class models: entities can be related by associations and may have attributes and methods. Therefore, *Entity*, *Attribute*, *Method*, *Association* and *AssociationEnd* are the possible protected resources. Figure 1 illustrates the metamodel of SecureUML+ComponentUML[†]. By using such SecureUML+ComponentUML[‡] it is possible, for instance, to specify the permissions a user playing a given role must have to execute a method (or to update an attribute) of a resource. In order to do so, it is necessary to instantiate the metaclasses *User*, *Role*, *Permission*, *ActionExecute*, *Method* (or *ActionUpdate*) and *Attribute*.

3 NormML: A Normative Modeling Language

NormML is a UML-based modeling language for the specification of norms in MAS. The choice for UML as metalanguage allows for an easy integration of NormML with UML-based MAS modeling languages such as AUML[19], AML[4] and MAS-

[†] The metamodel of SecureUML+ComponentUML (from now referred as SecureUML metamodel) is available at http://www.ic.uff.br/~viviane.silva/normML/secureUML.pdf

[‡] The metamodel of SecureUML+ComponentUML (from now referred as SecureUML metamodel) is available at http://www.ic.uff.br/~viviane.silva/normML/secureUML.pdf

ML[25]. Moreover, metamodel-based validation techniques may be applied to norms specified in NormML.

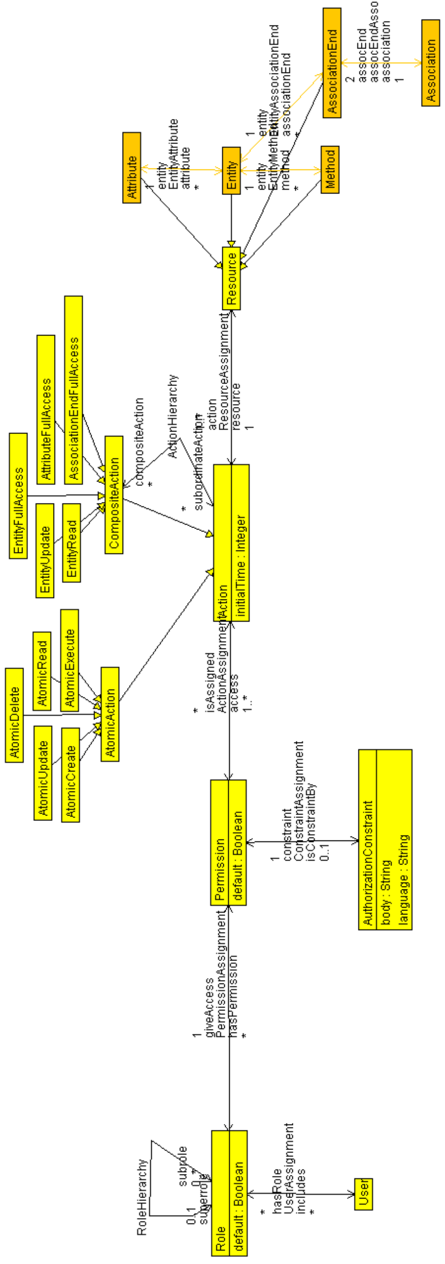


Figure 1. SecureUML+ComponenteUML metamodel

As mentioned in Section 2, NormML extends SecureUML modeling language. The NormML metamodel extends the SecureUML metamodel with the following basic elements: *Norm*, *Agent* and *AgentAction*. The NormML metamodel also includes a set of *invariants* that guarantees the well-formedness of a norm and several *operations* that are used to identify conflicts between two given norms.

3.1 The NormML Metamodel

The NormML metamodel extends the Secure UML metamodel in order to view norms as security policies, as mentioned in Section 2. While in Secure UML it is possible to define *permissions* a *user* has, i.e., the constraints that a user, in a given role, must fulfill to perform actions over the system resources, in NormML is possible to define the *norms* (obligations, permissions or prohibitions) an *entity* must obey, i.e., it is possible to describe the set of actions an agent, a role or an agent playing a role is obliged, permitted or prohibited to execute, conditioned by the execution of other actions. Figure 2 presents the NormML metamodel. (Some of SecureUML metaclasses are not presented for readability purposes.) A norm corresponds to an instance of the NormML metamodel, i.e., it is defined by instantiating several metaclasses and their relationships from the NormML metamodel. A norm may be either a permission (by instantiating the metaclass *NormPermission)*, a prohibition (by instantiating the metaclass *NormProhibition)* or an obligation (by instantiating the metaclass *NormObligation)*.

A norm may constraint the behavior of *Agents* by restricting the behavior of any given agent playing a given *Role*, or by restricting the behavior of a specific agent while playing a role. This is captured by the *Agent<->Role relationship*.

NormML inherits four resource kinds from SecureUML: *Attribute*, *Method*, *Entity* and *AssociationEnd*. It extends the set of resources with agent's actions and roles' actions represented by the metaclass *AgentAction*. Thus, it is possible to describe norms to control the access to attributes, methods, objects and association ends, and also to control the execution of the actions of agents and roles.

Each resource kind has a set of actions that can be used to control access to a resource. For instance, attributes are associated with the actions *read*, *update* and *full access (read+update)*. In the case of actions of agents and roles (*AgentAction* metaclass), the behavior that applies to it is the *execution* of the action.

Furthermore, NormML allows for the specification of the time period that a norm is *active*, which is represented by the metaclass *NormConstraint*. If a norm is conditioned by a *Before* clause, it means that the norm is active before the execution of the action(s) described in the *Before* clause. If a norm is conditioned by an *After* clause, it means that the norm is active only after the execution of the action(s) described in the *After* clause. In the case of a *Between* clause, the norm is only active during the period delimited by two groups of actions.

In order to illustrate the use of NormML to model the norms of a MAS, consider norms *N1*, *N2* and *N3* in Table 1. Figure 3, Figure 4 and Figure 5 illustrates the norm diagrams of *N1*, *N2* and *N3*, respectively.

N1: Seller is *obliged* to give the good to the buyer *after* the given buyer paid for it.

<u>*N2:*</u> Seller is *permitted* to update the price of a good *before* a buyer pays for it.
<u>*N3:*</u> Buyer is *prohibited* to return a good he/she has bought.

Table 1 - Norm example

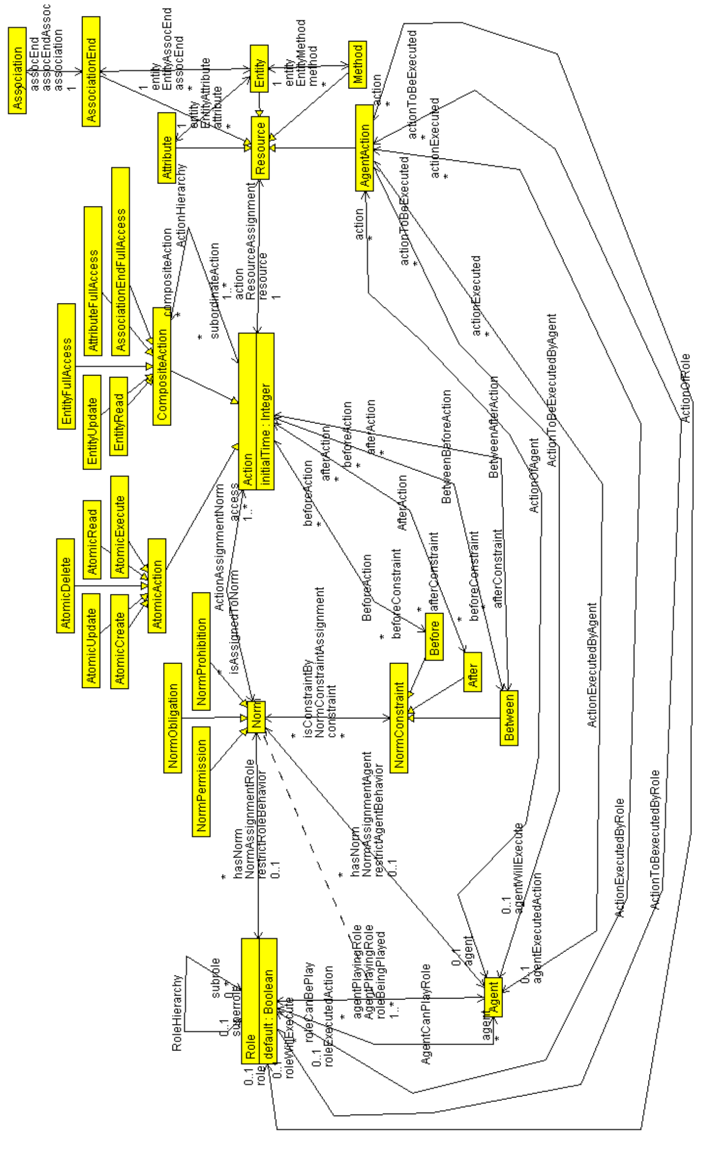


Figure 2. NormML **metamodel**

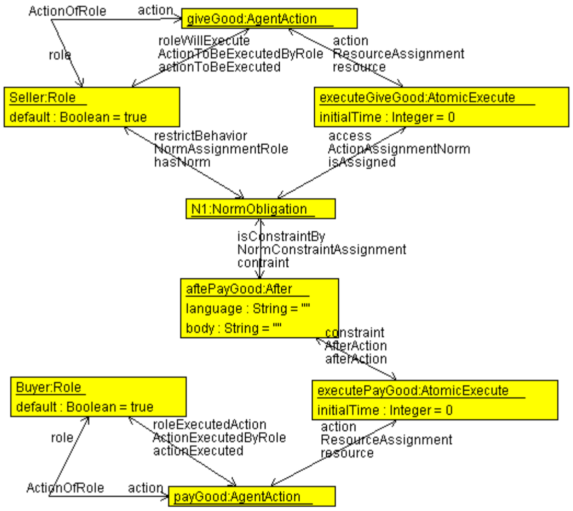


Figure 3. Norm N1 described by using NormML

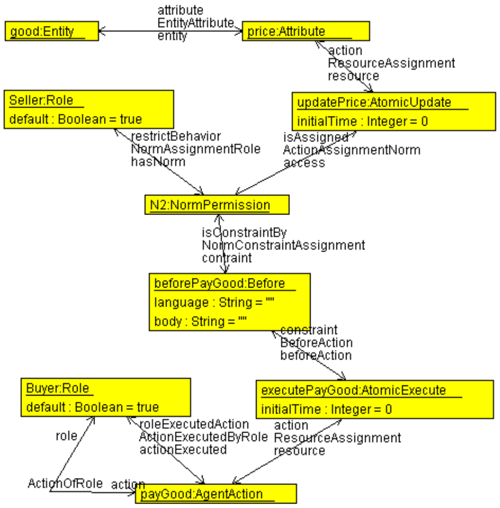


Figure 4. Norm N2 described by using NormML

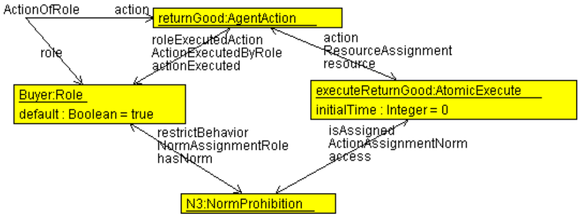


Figure 5. Norm N3 described by using NormML

3.2 Validating the Norms

The process of validating a norm encompasses two steps. First, the norm, as an instance of the NormML metamodel, is checked according to the invariants of the metamodel. The invariants check if the norm is well-formed according to the metamodel specification. The second step checks if any given two norms are in conflict.

Well-formed norms

Not all the norms that can be instantiated from the metamodel are well-formed. Examples of well-formed rules of the NormML metamodel are[§]:

WFR1: The resource AgentAction can only be linked with the atomic action called AtomicExecute. Any other atomic action does not apply to AgentAction.

```
context AgentAction
inv: AgentAction.allInstances-> forAll(aa|aa.action->
    select(a|not(a.oclIsTypeOf(AtomicExecution)))->isEmpty())
```

WFR2: The resource AgentAction cannot be constrained by Permission. Although the metaclass *Permission* is defined in the Secure UML metamodel to define the permissions a user has over resources, the resource *AgentAction* can only be used by *Norms* to restrict the actions of an agent.

```
context Permission
inv: Permission.allInstances->forAll(p|p.accesses->
   select(a|a.resource.oclIsTypeOf(AgentAction))->isEmpty())
```

WFR3: A norm that regulates the execution of a given action cannot be conditioned by the execution of the same action by the same agent. An agent cannot be obliged, permitted or prohibited to execute an action conditioned to the execution of such action. This rule uses four operations in order to guarantee that the action being regulated by the norm is not in the set of actions of the *Before*, *After* or *Between* constraints.

[§] Some of the well-formed rules of the NormML metamodel are available in http://www.ic.uff.br/~viviane.silva/normML/normML.pdf. We are using OCL [17] to describe the well-formed rules and also the operations to check conflicts.

```
context Norm
inv: self.GetAgentExecutedActionInBeforeConstraint->
         union(self.GetAgentExecutedActionInAfterConstraint)->
         union(self.GetAgentExecutedActionInBetweenConstraint)->
         excludes(self.GetAgentExecutedActionRestrictedByNorm)
```

Checking for Conflicts

After verifying the well-formedness of the norms, it is important to check if there are conflicts between the norms. Two norms are in conflict, or are incompatible, if:

1. One states a permission and another one a prohibition to execute the same action and such norms are active during the same period of time or during periods of time that intersects. The conflict occurs because the agent is permitted and prohibited to execute an action at the same time. Example:

N3a: Buyer is *prohibited* to return a good it has bought.

N3b: Buyer is *permitted* to return a good it has bought *before* using it.

The activation time of N3a and N3b intersects since N3a states an unlimited prohibition. Thus, these norms are in conflict.

2. One norm states an obligation and another one a prohibition over the same action and such norms are active during the same period of time or during periods of time that intersect. The conflict occurs because the agent is obliged and prohibited to execute an action at the same time. Example:

N1a: Seller is *obliged* to give the good to buyer *after* the given buyer paid for it.

N1b: Seller is *prohibited* to give the good to buyer *before* the latter pays for it.

The activation time of N1a and N1b do not intersect. These norms are not in conflict since the seller is not being obliged and prohibited to execute the same action during the same period of time.

3. One norm states a permission and another one an obligation over the same action and such norms are not active during the same period of time. A conflict may occur if an agent is obliged to execute an action that it is not permitted to. Example:

N2a: Seller is *permitted* to update the price of a good *before* a buyer pays for it.

N2b: Seller is *obliged* to update the price of a good *after* a buyer pays for it.

The activation time of N2a and N2b do not intersect, thus these norms are in conflict.

In addition, we also consider that a conflict can be caused due to the relationship between an agent and the roles it is playing.

- *A norm applied to a role and another one applied to an agent may be in conflict:* A norm applied to a role restricts the behavior of all agents playing such role. Therefore, when searching for conflicts, it is important to check the incompatible norms that are applied to roles and also the ones applied to agents that are able to play such roles. Note that agents can play several roles.
- *A norm applied to a role and another one applied to an agent playing the role may be in conflict:* Since the norm applied to a role regulates the behavior of all agents applying such role, when searching for conflicts, it is important to check the incompatible norms that are applied to roles and to agents playing roles.
- *A norm applied to an agent and another one applied to the agent playing a role may be in conflict:* Since both norms will regulate the behavior of the same

agent, when searching for conflicts it is important to check the incompatible norms that are applied to agents and to agents playing roles.

Note that two norms applied to different roles are never in conflict even though the same agent can play both roles. Although an agent can play more than one role at the same time, an action is always executed in the context of one role. We understand that an agent must be able to obey each norm separately while playing the roles.

The operation *CheckConflict* illustrated below should be used to check conflicts between two norms. First, it checks if the norms are the same and, it they are not, if they apply to the same or related entities (as described above). Then, three important auxiliary operations[**] are used to check conflicts between an obligation and a prohibition, between an obligation and a permission and between a permission and an obligation.

```
context :: CheckConflict(norm1,norm2) : String
body if ( (norm1<>norm2)
then(
 if (CheckSameOrRelatedEntities(norm1,norm2)
 then(
  if (CheckConflictObligationProhibition(norm1, norm2) =“conflict” OR
      CheckConflictObligationPermission (norm1, norm2) =“conflict” OR
      CheckConflictPermissionObligation (norm1, norm2) =  “conflict”)
  then (“conflict”)
  else(
   if (CheckConflictObligationProhibition(norm1,norm2) = “conflictFree” AND
       CheckConflictObligationPermission(norm1,norm2) = “conflictFree” AND
      CheckConflictPermissionObligation (norm1, norm2) = “conflictFree”)
    then( “conflictFree”)
    else (“cannotBeVerified”)
  ))
 else (“conflictFree”))
else (“sameNorm”)
```

In order to exemplify one of the three auxiliary operations, let’s focus on the *CheckConflictObligationPermission* operation, since it is frequently forgotten by other authors. First, this operation checks if it is dealing with a permission and an obligation and if both norms regulate the same actions (*case 0* in operation *CheckConflictObligationPermission*). Second, it checks if the permission is not conditioned to any situation (*case 1*). In such case, there is not a conflict because the entity is always permitted to execute the action it is being obliged.

Then, it checks if the norms are constrained to the same set of constraints, i.e., if the actions that activate and deactivate the norms are the same (*case 2*). If it is the case that both norms are constrained by a *Before* clause, then there is not a conflict since the entity is being obligated to execute an action while it is permitted to. *Cases 2.2, 2.3* and *2.4* in operation *CheckConflictObligationPermission* are similar. However, if the obligation is conditioned by a *Between* clause and the permission to a *Before* or an *After* (*cases 2.5* and *2.6*) it is not possible to conclude during design time if these norms are in conflict. It will depend on the sequence of the executions of the actions that will activate the norms. On the other hand, if the permission is being constrained to a *Between* condition and the obligation by a *Before* or an *After* (*cases*

[**] The implementation of such operations can be found in http://maude.sip.ucm.es/~viviane/normML.txt

2.7 and *2.8*), both norms are in conflicts since the entity is being obliged to execute an action without permission.

If the norms are not conditioned by the same set of conditions, then it is only possible to affirm that they are in conflict (i) in the case one of the norms is conditioned to an *After*[††] and the other to a *Before*[‡‡] (*cases 3.1* and *3.2*) and (ii) in the case the permission is conditioned to a *Between* and the obligation to a *Before* (*case 3,3*). In both cases the agent is being obliged to execute a norm that it is not permitted to.

```
context :: CheckConflictObligationPermission(norm1,norm2) : String
body
if ((norm1.oclIsTypeOf(NormObligation) and norm2.oclIsTypeOf(NormPermission))
 or (norm1.oclIsTypeOf(NormPermission) and norm2.oclIsTypeOf(NormObligation)))
then ( **case 0,check if norms applies to same action**
 if (norm1.accesses=norm2.accesses
 then (  **case 1**
   if ((norm1.oclIsTypeOf(NormPermission) and
        norm1.ActionsInConstraintOfNorm()->isEmpty()() ) or
       (norm2.oclIsTypeOf(NormPermission) and
        norm2.ActionsInConstraintOfNorm()->isEmpty()()))
   then ( “conflictFree” )
   else ( **case 2**
     if (CheckSameSetOfConstraint(norm1,norm2)
     then ( **case 2.1**
       if (CheckBeforeBeforeNorms(norm1,norm2))
       then ( “conflicFree”)
       else ( **case 2.2**
         if (CheckAfterAftertNorms(norm1,norm2))
         then ( “conflictFree” )
         else ( **case 2.3**
           if (CheckBetweenBetweentNorms(norm1,norm2))
           then ( “conflictFree” )
           else ( **case 2.4**
             if (CheckAfterBeforeAfterBeforeNorms(norm1,norm2))
             then ( “conflictFree” )
             else ( **case 2.5**
            if(CheckBetweenObligationBeforePermissionNorms(norm1,norm2))
               then ( “cannotBeVerified” )
               else ( **case 2.6**
                 if (CheckBetweenObligationAfterPermissionNorms(norm1,norm2))
                 then ( “cannotBeVerified” )
                 else ( **case 2.7**
                   if (CheckBetweenPermissionBeforeObligation(norm1,norm2))
                   then ( “conflict” )
                   else ( **case 2.8**
                     if (CheckBetweenPermissionAfterObligationNorms
                                                                 (norm1,norm2))
                     then ( “conflict” )
                     else ( “cannoBeVerified” )
       ))))))) )
     else ( **case 3**
       **case 3.1**
       if (CheckBeforePermissionAfterObligationNorms(norm1,norm2))
       then ( “conflict” )
```

[††] Note that we are considering that the *After* condition specifies that the norm is only valid when all the actions identified in the condition are executed.

[‡‡] Note that we are considering that the *Before* condition specifies that the norm is only valid while none of the actions described in such condition is executed.

```
       else  ( **case 3.2**
         if (CheckAfterPermissionBeforeObligationNorms(norm1,norm2))
         then ( “conflict” )
         else  ( **case 3.3**
           if (CheckBetweenPermissionBeforeObligationNorms(norm1,norm2))
           then ( “conflict” )
           else  ( “cannotBeVerified” )
   )))) )
 else ( “conflictFree”  ))
else ( “conflictFree” )
```

3.3 The Use of MOVA to Model, Validate and Query the Norms

MOVA (Modeling and Validation group) tool [6] was used as a modeling tool (i) to describe the NormML metamodel, (ii) to create the normative models, (iii) to check the well-formedness of the norms and their conflicts, and also (iv) to inspect the normative models. MOVA allows for the creation of class diagrams, the definition of a set of invariants and operations over such diagrams and checking if object diagrams respect the invariants defined in class diagrams. We used MOVA to define the NormML *metamodel* as a class diagram and to describe the *well-formed rules* of the metamodel as invariants of the class diagram. The *normative models* were then described as object diagrams and checked if they comply with these invariants.

By using MOVA it is also possible to query the object diagram (i.e., to define queries over such models) that can use operations defined in the class diagram. Such mechanism was used not only to *check the conflicts* between the modeled norms by using the operations that investigate the possible conflicts but also to *explore the normative models* themselves. Such investigation is fundamental when dealing with large-scale MAS that typically define a large number of norms. After describing hundreds norms it is almost impossible to find out, without the helping of a tool, all the norms applied to a role, for instance.

4 RELATED WORK

In this section we briefly describe how some methodologies and modeling languages deal with the modeling of the system norms. In addition we also present and compare works that have also proposed approaches to deal with norm conflicts.

4.1 Methodologies

Methodologies such as MESSAGE [3], Tropos [2] and Prometheus [22] do not address the problem of identifying and explicitly modeling norms or organizational rules. However, others such as Gaia, SODA, MaSE and PASSI state the importance of modeling organization rules during the analysis and design phases.

Gaia affirms that the explicit identification of such rules in the analysis phase is very important for the correct understanding of the characteristics that the organization-to-be must express and for the subsequent definition of the system structure by the designer. Although they have proposed a formal language to model the norms, they have not described any mechanism to validate the norms in order to find out conflicts and to verify if the elements being referred to by the norms are elements being modeled.

In [7] the authors propose the integration of organizational rules into the MaSE methodology by extending its analysis and design phases. The rules are modeled in the analysis phase, while in the design phase, the organization tasks related to the implementation and enforcement of those rules are described. Like Gaia, MaSE defines a formal language for describing norms but have not proposed how to find out norms' conflicts or how to check consistency between the elements described in the norms and the elements being modeled.

SODA states the need for modeling social rules as agents' interactions in the analysis phase and defines social models expressive enough to model the society interaction rules in the design phase. However, as opposed to Gaia and MaSE, this methodology neither presents a guideline to define such rules nor describes in details the characteristics of the proposed social models.

In the role description phase of the PASSI methodology, it is possible to introduce social rules (or organization rules) in the UML class diagrams used to model the agents, their roles and actions. The rules may be expressed in OCL or other formal, or semi-formal manner depending on one's needs. The two main drawbacks of this approach to model norms are: (i) there is not a method to verify if the elements being described in the norms are modeled in the system diagrams; and (ii) they do not propose any mechanism to check if the norms have conflicts.

4.2 Modeling Languages

Both AUML and AML recognize the need for modeling norms but have not defined any modeling technique to describe them. AML states that roles are used to define a normative behavioral repertoire of entities but has not proposed the modeling of norms. Thus, it is not possible to point out the permissions, obligations and prohibitions of an agent playing a role.

In AOR [28] the use of deontic logic to describe norms is still under investigation. Although it is possible to describe rights (or permissions) and prohibitions, it is still not possible to describe obligation. In addition, there is not any mechanism to detect norms conflicts, even though it is possible to describe them.

MAS-ML originally proposed the modeling of duties (or obligations) and rights as actions associated with roles. However, it is not possible to model more complex norms such as the ones conditioned to an event or to check their conflicts.

4.3 Other Approaches that deal with Norm Conflicts

There are several works that introduce approaches to check conflicts between norms and to solve such conflicts [9][13][21][23][26]. Since we have not presented any suggestion to the resolution of conflicts, we compare our approach with the ones that can find out the conflicts.

In [23] the authors identify three forms of conflict/inconsistency called total-total, total-partial and intersection. The approach we propose to validate the set of norms and detect conflicts can capture these three forms of conflict/inconsistency. In [9] several aspects of some types of conflicts and the problems they arise are discussed. In particular, the authors discuss the difference between deontic inconsistencies, which occur when actions are simultaneously prohibited and permitted, and deontic conflicts, which occur when actions are simultaneously prohibited and obliged. In our approach we present solutions to deal with these two types of conflicts.

The model presented in [14], called NoA, is able to detect conflicts between norms at runtime and propose resolutions to those conflicts. They state that by allowing conflicts it has partial benefits in the engineering of multi-agent systems. Thus, the main difference between their approach and ours is that our mechanism must be used to check norms at design time. In our point of view, at least the norms defined by the design must be conflict-free before the execution of the system.

Differently from us, in [12] the authors present an approach to detect conflict based on the time a norm is activated. In our approach we have not associated a norm with an activation time but with the execution of a set of actions that activates the norm. In [26] the authors present an approach to detect conflicts between related norms, i.e., norms applied to the same agent/role, restricting the same actions and whose activation periods overlap. The mechanism used to detect conflicts proposed in our paper is based on the approach presented in [26]. We extend such approach to consider conflicts between norms that state permissions and obligations—the authors in [26] only consider permissions and prohibitions or obligations and prohibitions—and to deal with activation time that is related to the execution of actions—the activation time proposed in [26] is related to values associated with attributes.

5 CONCLUSIONS AND FUTURE WORK

We have presented NormML, a normative modeling language that builds on role-based access control concepts. By using NormML it is possible to identify roles, agents and actions of a system while modeling its norms. Since NormML is based on UML, the integration of such language with any multi-agent system modeling language also based in UML, such as AUML, AML and MAS-ML, is facilitated. The roles, agents and actions identified while modeling the norms must be modeled in the agent-oriented models provided by such modeling languages.

We have defined a set of invariants and operations that makes possible the validation of the norms by verifying their well-formedness and by checking the possible conflicts between norms. We have defined three main operations to detect conflicts between an obligation and a permission, an obligation and a prohibition, and a permission and a prohibition.

We are in the process of extending the language to describe temporal restrictions and also sanctions. In order to be able to define the NormML metamodel, we have based such definition in the normative grammar proposed in [24]. This grammar extends the normative language proposed by Garcia-Camino et al. [10] with the notion of non-dialogical actions proposed by Vazquez-Salceda et al. [27] and with the definition of sanctions and relationships between norms stated by Lopez y Lopez et al. in [11][16]. However, the current version of NormML does not contemplate the definition of sanctions or temporal conditions.

It is also our intension to define a sequence diagram for NormML to describe the sequence of the executed actions. By using such diagram it will be possible to check conflicts that depend on the sequence of the executed actions (as mentioned in Section 3.2) and it will also be possible to identify the norms that are active and the ones that were violated.

REFERENCES

[1] Basin, D. A., Doser, J. and Lodderstedt, T. 2006. Model driven security: From UML models to access control infrastructures. ACMTrans. on Soft. Eng. and Met.15(1)pp.39-91

[2] Bresciani, P, Perini, A., Giorgini, P., Giunchiglia, F., Mylopoulos, J. 2004. Tropos: An Agent-Oriented Software Development Methodology. In JAAMAS, 8, pp. 203-236.

[3] Caire, G., Coulier, W., Garijo, F., Gomez, J., Pavon, J., Leal, F., Chainho, P., Kearney, P., Stark, J., Evans, R., Massonet, P. 2002. Agent Oriented Analysis Using Message/UML. In Agent-Oriented Software Engineering II, LNCS 2222, pp. 119-135.

[4] Caronervenka, R., Trenccaronansky, I., Calisti M. and Greenwood, D. 2005. AML: Agent Modeling Language Toward Industry-Grade Agent-Based Modeling. InAgent-Oriented Software Engineering V, LNCS 3382, pp. 31-46.

[5] Ciancarini, P., Omicini, A., and Zambonelli, F. 2000. Multiagent System Engineering: the Coordination Viewpoint. In Intelligent Agents VI., LNAI 1767, pp. 250-259.

[6] Clavel, M., Egea, M., Silva, V. 2007. The MOVA Tool: A Rewriting-Based UML Modeling, Measuring, and Validation Tool. In Proc. Workshop de Demonstraciones de Herramientas de las Jornadas de Ingeniería del Software y Bases de Datos, pp. 393-394.

[7] Cossentino, M. 2005. From requirements to code with the PASSI methodology. In Agent-oriented Methods, Idea group, pp. 79-106.

[8] DeLoach, S. 2002. Modeling Organizational Rules in the Multiagent System Engineering Methodology", in Proc. Canadian Conf.on Artificial Intelligence, LNAI 2338, pp. 1-15.

[9] Elhag, A; Breuker, J.; Brouwer P. 2000. On the formal analysis of normative conflicts. Information and Communication Technology Law, 9(3), pp. 2007-217.

[10] Ferraiolo, D. F., Kuhn, D. R., and Chandramouli, R. 2007. Role-Based Access Control. Artech House Publishers, 2nd Edition.

[11] García-Camino, A., Noriega, P. and Rodríguez-Aguilar, J. 2005. Implementing Norms in Electronic Institutions. In Proc. of Autonomous Agents and MAS, ACMPress,pp.667-673.

[12] García-Camino, A., Noriega, P. and Rodríguez-Aguilar, J. 2007. An algorithm for conflict resolution in regulated compound activities. In Engineering Societies in the Agents World VII, LNCS 4457, Spriger-Verlag, pp.193-208.

[13] Kagal, L; Finin, T. (2007). Modeling conversation policies using permissions and obligations. Journal of Autonomous Agents and Multagent Systems, 14(2), pp. 187-206.

[14] Kollingbaum, M; Norman, T.; Preece, A; Sleeman, D. 2007. Norm Conflicts and Inconsistencies in Virtual Organisations. In Coordination, Organizations, Institutions, and Norms in Agent Systems II, LNCS, 4386.

[15] López, F. 2003. Social Power and Norms: Impact on agent behavior. PhD thesis, Univ. of Southampton, Faculty of Eng. and Applied Science, Depart. Electronics and Computer Science.

[16] López, F. Luck, M. and d'Inverno, M. 2002. Constraining autonomy through norms. In Proceedings of Autonomous Agents and Multi-Agent Systems, ACM Press, pp. 674-681

[17] Object Management group, OCL Specification, OMG. Available in http://www.omg.org/docs/ptc/03-10-14.pdf.

[18] Object Management group, UML 2.0, OMG. Available in http://www.uml.org/.

[19] Odell, J., Parunak, H., and Bauer, B. 2000. Extending UML for Agents. In Proc. Agent-Oriented Information Systems Workshop at National Conf. of AI, pp. 3-17.

[20] Omicini, A. 2002. SODA: Societies and Infrastructures in the Analysis and Design of Agent-Based Systems. In Agent-Oriented Software Engineering, LNCS1957, pp. 311-326.

[21] Oren, N.; Luck, M; Miles, S., Norman, T. 2008. An argumentation-inspired heuristic for resolving normative conflict. In Proceedings of the 5th Workshop on Coordination, Organizations, Institutions and Norms in Agent Systems.

[22] Padgham, L. and Winikoff, M. 2002. Prometheus: A Methodology for Developing Intelligent Agents. In Proc.Agent-Oriented Software Engineering Workshop, pp. 174-185.

[23] Ross, A. 1958. On Law and Justice. Stevens & Sons.

[24] Silva, V. 2008. From the Specification to the Implementation of Norms: An Automatic Approach to Generate Rules from Norms to Govern the Behaviour of Agents. JAAMAS, Special Issue on Norms in Muli-Agent Systems, volume 17, number1, pp. 113-155.

[25] Silva, V.; Lucena, C. 2004. From a Conceptual Framework for Agents and Objects to a Multi-Agent System Modeling Language. JAAMAS, 9(1-2), Kluwer, pp. 145-189.

[26] Vasconcelos, W., Kollingbaum, M., Norman, T. 2007. Resolving Conflict and Inconsistency in Norm-Regulated Virtual Organizations. In Proc. AAMAS.

[27] Vázquez-Salceda, J., Aldewereld, H., Dignum, F. 2004. Implementing Norms in Multiagent Systems. In LNAI 3187, pp. 313-327.

[28] Wagner, G. 2003. The Agent-Object-Relationship Metamodel: Towards a Unified View of State and Behaviour. In Information Systems, vol. 28(5).

[29] Zambonelli, F., Jennings, N., Wooldridge, M. 2003. Developing Multiagent Systems: The Gaia Methodology. ACM Trans. on Soft. Eng. and Methodology, Vol., no 3, pp. 317-370.

Monitoring Social Expectations in Second Life

Stephen Cranefield and Guannan Li

Department of Information Science
University of Otago
PO Box 56, Dunedin 9054, New Zealand
scranefield@infoscience.otago.ac.nz

Abstract. Online virtual worlds such as Second Life provide a rich medium for unstructured human interaction in a shared simulated 3D environment. However, many human interactions take place in a structured social context where participants play particular roles and are subject to expectations governing their behaviour, and current virtual worlds do not provide any support for this type of interaction. There is therefore an opportunity to adapt the tools developed in the MAS community for structured social interactions between software agents (inspired by human society) and adapt these for use with the computer-mediated human communication provided by virtual worlds.

This paper describes the application of one such tool for use with Second Life. A model checker for online monitoring of social expectations defined in temporal logic has been integrated with Second Life, allowing users to be notified when their expectations of others have been fulfilled or violated. Avatar actions in the virtual world are detected by a script, encoded as propositions and sent to the model checker, along with the social expectation rules to be monitored. Notifications of expectation fulfilment and violation are returned to the script to be displayed to the user. This utility of this tool is reliant on the ability of the Linden scripting language (LSL) to detect events of significance in the application domain, and a discussion is presented on how a range of monitored structured social scenarios could be realised despite the limitations of LSL.

1 Introduction

Much of the research in multi-agent systems addresses techniques for modelling, constructing and controlling open systems of autonomous agents. These agents are taken to be self-interested or representing self-interested people or organisations, and thus no assumptions can be made about their conformance to the design goals, social conventions or regulations governing the societies in which they participate. Inspired by human society, MAS researchers have adopted, formalised and created computational infrastructure allowing concepts from human society such as trust, reputation, expectation, commitment and narrative to be explicitly modelled and manipulated in order to increase agents' awareness of the social context of their interactions. This awareness helps agents to carry out their interactions efficiently and helps preserve order in the society, e.g. the existence of reputation, recommendation and/or sanction mechanisms discourages anti-social behaviour.

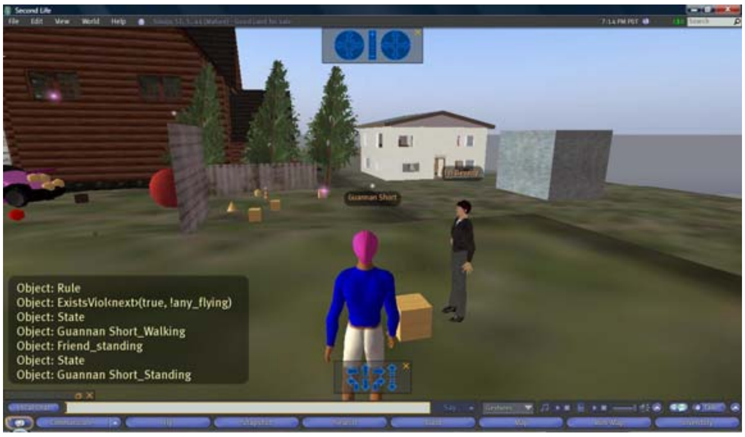
Dagstuhl Seminar Proceedings 09121
Normative Multi-Agent Systems
http://drops.dagstuhl.de/opus/volltexte/2009/1906

As the new ‘Web 2.0’ style Web sites and applications proliferate, people’s use of the Web is moving from passive information consumption to active information sharing and interaction within virtual communities; in other words, for millions of users, the Web is now a place for social interaction. However, while Web 2.0 applications provide the middleware to enable interaction, they generally provide no support for users to maintain an awareness of the social context of their interactions (other than basic presence information indicating which users in a ‘buddy list’ online). There is therefore an opportunity for the software techniques developed in MAS research for maintaining social awareness to be applied in the context of electronically mediated human interaction, as well as in their original context of software agent interaction.

This paper reports on an investigation into the use of one such social awareness tool in conjunction with the Second Life online virtual world. Second Life is a ‘Web 3D’ application providing a simulated three dimensional environment in which users can move around and interact with other users and simulated objects [1]. Users are represented in the virtual world by animated avatars that they control via the Second Life Viewer client software. Human interaction in virtual worlds is essentially unconstrained—the users can do whatever they like, subject to the artificial physics of the simulated world and a few constraints that the worlds support, such as the ability of land owners to control who can access their land. However, many human interactions take place in a structured social context where participants play particular roles and there are constraints imposed by the social or organisational context, e.g. participants in a meeting should not leave without formally excusing themselves, and students in an in-world lecture should remain quiet until the end of the lecture. Researchers in the field of multi-agent systems have proposed (based on human society) that the violation of social norms such as these can be discouraged by publishing explicit formal definitions of the norms, building tools that track (relevant) events and detect any violations, and punishing offenders by lowering their reputations or sanctioning them in some other way [2]. Integrating this type of tool with virtual worlds could enhance the support provided by those worlds for social activities that are subject to norms.

In this research we have investigated the use of a tool for online monitoring of ‘social expectations’ [3] in conjunction with Second Life. The mechanism involves a script running in Second Life that is configured to detect and record particular events of interest for a given scenario, and to model these as a sequence of state descriptions that are sent to an external monitor along with a property to be monitored. The monitor sends notifications back to the script when the property is satisfied so that the user can be informed.

The rest of this paper is structured as follows. Section 2 describes how we have used the Linden Scripting Language to detect avatars in Second Life and create a sequence of propositional state models to send to the monitor. The architecture for communication between this script and the monitor is presented in Section 3. Section 4 discusses the concept of conditional social expectations used in this work, and the model checking tool that is used as the expectation monitor. Section 5 presents some simple scenarios of activities in Second Life being monitored, and Section 6 discusses some issues arising from limitations of the Linden Scripting Language and the temporal logic used to



```
Object: Rule
Object: ExistsViol<next>(true, !any_flying)
Object: State
Object: Guannan Short_Walking
Object: Friend_standing
Object: State
Object: Guannan Short_Standing
```

Fig. 1. The Second Life Viewer

express rules. Some related work is described in Section 7, and Section 8 concludes the paper.

2 Detecting events in Second Life

As shown in Figure 1, the Second Life Viewer provides, by default, a graphical view of the user's avatar and other objects and avatars within the view. The user can control the 'camera' to obtain other views. Avatars can be controlled to perform a range of basic animations such as standing, walking and flying, or predefined "gestures" that are combinations of animation, text chat and sounds. Communication with other avatars (and hence their users) is via text chat, private instant messages, or audio streaming. The user experience is therefore a rich multimedia one in which human perception and intelligence is needed to interpret the full stream of incoming data. However, the Linden Scripting Language (LSL [4]) can be used to attach scripts to objects (e.g. to animate doors), and there are a number of sensor functions available to detect objects and events in the environment. These scripts are run within the Second Life servers, but have some limited ability to communicate with the outside world.

LSL is based on a state-event model, and a script consists of defined states and handlers for events that it is programmed to handle. Certain events in the environment automatically trigger events on a script attached to an object. These include collisions with other objects and with the 'land', 'touches' (when a user clicks on the object), and money (in Linden dollars) being given to the object. Some other types of event must be explicitly subscribed to by calling functions such as `llSensor` and `llSensorRepeat` for scanning for avatars and objects within a given arc and range, `llListen` for detecting chat messages from objects or avatars within hearing range, and `llSetTimerEvent` for setting a timer. These functions take parameters that provide some selectivity over what is sensed, e.g. a particular avatar name or object type can be specified in `llListen`, and `llListen` can be set to listen on a particular channel, for a message from a particular avatar, and even for a particular message.

In this paper we focus on the detection of other avatars via the function `llSensorRepeat`, which repeatedly polls for nearby avatars (we choose not to scan for objects also) at an interval specified in a parameter. A series of `sensor` events are then generated, which indicate the number of avatars detected in each sensing operation. A loop is used to get the unique key that identifies each of these avatars (via function `llDetectedKey`) and the avatar's name (via `llDetectedName`). The key can then be used to obtain each avatar's current basic animation (via `llGetAnimation`). The script can be configured with a filter list specifying which avatar/animation observations should be either recorded or ignored, where the specified avatar and animation can refer to a particular value, or "any". Detected avatar animations are filtered through this list sequentially, resulting in a set of $(avatar\_name, animation)$ pairs that comprise a model of the current state of the avatars within sensor range. Another configuration list specifies the optional assignment of avatars to named groups or roles such as "Friend" or "ClubOfficial". There is currently no connection with the official Second Life concept of a user group (although official group membership can be detected). Group names can also be included in the filter list, with an intended existential meaning, i.e. a pair $(group\_name, animation)$ represents an observation that *some* member of the group is performing the specified animation. The configuration lists provide scenario-specific relevance criteria on the observed events, and are read from a 'notecard' (a type of avatar inventory item that is commonly used to store textual configuration data for scripts), along with the property to be monitored.

When the script starts up, it sends the property to be monitored to the monitor. It then sends a series of state descriptions to the monitor as sensor events occur. However, we choose not to send a state description if there is no change since the previous state, so states represent periods of unchanging behaviour rather than regularly spaced points in time. State descriptions are sets of proposition symbols of the form *avatar_animation* or *group_animation*.

This process can easily be extended to handle other types of Second Life events that have an obvious translation to propositional (rather than predicate) logic, such as detecting that an avatar has sent a chat message (if it is not required to model the contents of the message). Section 6 discusses this further.

3 Communication between Second Life and the monitor

Second Life provides three mechanisms for communication with entities outside their own server or the Second Life Viewer: scripts can send email messages, initiate HTTP requests, or listen for incoming XML-RPC connections (which must include a parameter giving the key for a channel previously created by the script). To push property and state information to the monitor we use HTTP. However, instead of directly embedding the monitor in an HTTP server, to avoid local firewall restrictions we have chosen to use Twitter [5] as a message channel. An XML-RPC channel key, the property to be monitored and a series of state descriptions are sent to a predefined Twitter account as direct messages using the HTTP API[1]. The Twitter API requires authentication, which can be achieved from LSL only by including the username and password in the URL in the form http://username:password@. . . ..

The monitor is wrapped by a Java client that polls Twitter (using the Twitter4J library [7]) to retrieve direct messages for the predetermined account. These are ignored until a pair of messages containing an XML-RPC channel key and a property to be monitored (prefixed with "C:" and "P:" respectively) are received, which indicates that a new monitoring session has begun. The monitoring session then consists of a series of messages beginning with "S:", each containing a list of propositions describing a new state. The monitor does not currently work in an incremental 'online' mode—it must be given a complete history of states and restarted each time a new state is received[2]; therefore, the Java wrapper must record the history of states. It also generates a unique name for each state (which the monitor requires).

Each time a state is received, the monitor (which is implemented in C) is invoked using the Java Native Interface (JNI). The rule and state history are written to files and the names passed as command-line arguments. An additional argument indicates the desired name of the output file. The output is parsed and, if the property is determined to be true in any state, that information is sent directly back to the Second Life script via XML-RPC.

Figure 2 gives an overview of the communication architecture.

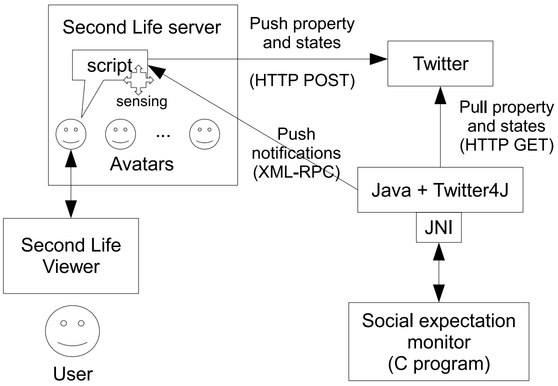


Fig. 2. The communications architecture

[1] Twitter messages are restricted to 140 characters and calls to the Twitter API are subject to a limit of 70 requests per hour, which is sufficient for testing our mechanism. For production use an alternative HTTP-accessible messaging service could be used, such as the Amazon Simple Queue Service [6].

[2] Work is in progress to add an online mode to the monitor.

4 Monitoring social expectations

4.1 Modelling social expectations

MAS researchers working on normative systems and electronic institutions [2] have proposed various languages for modelling the rules governing agent interaction in open societies, including abductive logic programming rules [8], enhanced finite state machine style models, [9], deontic logic [10], and institutional action description languages based using formalisms such as the event calculus [11].

The monitor used in this work is designed to track rules of *social expectation*. These are temporal logic rules that are triggered by conditions on the past and present, resulting in *expectations* on present and future events. The language does not include deontic concepts such as obligation and permission, but it allows the expression of social rules that impose complex temporal constraints on future behaviour, in contrast to the simple deadlines supported by most normative languages. It can also be used to express rules of social interaction that are less authoritative than centrally established norms, e.g. conditional rules of expectation that an agent has established as its personal norms, or rules expressing learned regularities in the patterns of other agents' behaviour. The key distinction between these cases is the process that creates the rules, and how agents react to detected fulfilments and violations.

Expectations become active when their condition evaluates to true in the current state. These expectations are then considered to be fulfilled or violated if they evaluate to true in a state without considering any future states that might be available in the model[3]. If an active expectation is not fulfilled or violated in a given state, then it remains active in the following state, but in a "progressed" form. Formula progression involves partially evaluating the formula in terms of the current state and re-expressing it from the viewpoint of the next state [12]. A detailed explanation is beyond the scope of this paper, but a simple example is that an expectation $\bigcirc\phi$ (meaning that ϕ must be true in the state that follows) progresses to the expectation ϕ in the next state.

4.2 The social expectation monitor

The monitoring tool we have used is an extension [3] of a model checker for hybrid temporal logics [13]. Model checking is the computational process of evaluating whether a formal model of a process, usually modelled as a Kripke structure (a form of nondeterministic finite state machine), satisfies a given property, usually expressed in temporal logic. For monitoring social expectations in an open system, we cannot assume that we can obtain the specifications or code of all participating agents to form our model. Instead our model is the sequence of system states recorded by a particular observer, in other words, we are addressing the problem of *model checking a path* [14]. The task of the model checker is therefore not to check that the overall system *necessarily* satisfies

[3] This restriction is necessary, for example, when examing an audit trail to find violations of triggered rules in *any* state. The standard temporal logic semantics would conclude that an expectation "eventually p" is fulfilled in a state s even if p doesn't become true until some later state s'.

a given property, but just that the observed behaviour of the system has, to date, satisfied it. The properties we use are assertions that a social expectation exists or has been fulfilled or violated, based on a conditional rule of expectation, expressed in temporal logic.

The basic logic used includes these types of expression, in addition to the standard Boolean constants and connectives (true, false, $\wedge$, $\vee$ and $\neg$):

- Proposition symbols. In our application these represent observations made in Second Life, e.g. *avatar_name_*sitting.
- $\bigcirc\phi$: formula ϕ is true when evaluated in the next state
- $\Diamond\phi$: ϕ is true in the current or some future state
- $\Box\phi$: ϕ is true in all states from now onwards
- $\phi \, \mathsf{U} \, \psi$: ψ is true at the current or some future state, and ϕ is true for all states from now until just before that state

$\Diamond$ and $\Box$ can be expressed in terms of U and are abbreviations of longer expressions.

The logic also has some features of Hybrid Logic [15], but these are not used in this work except for the use of a *nominal* (a proposition that is true in a unique state) in the output from the model checker to ‘name’ the state in which a fulfilled or violated rule of expectation became active.

Finally, the logic includes the following operators related to conditional rules of expectation, and these are the types of expression sent from the Second Life script to the model checker:

- $\mathsf{ExistsExp}(Condition, Expectation)$
- $\mathsf{ExistsFulf}(Condition, Expectation)$
- $\mathsf{ExistsViol}(Condition, Expectation)$

where $Condition$ and $Expectation$ can be any formula that does not include ExistsExp, ExistsFulf and ExistsViol.

The first of these operators evaluates to true if there is an expectation existing in the current state that results from the rule specified in the arguments being triggered in the present or past. The other two operators evaluate to true if there is currently a fulfilled or violated expectation (respectively) resulting from the rule.

Formal semantics for this logic can be found elsewhere [3].

The input syntax to the model checker is slightly more verbose than that shown above. In particular, temporal operators must indicate the name of the ”next state modality” as it appears in the input Kripke structure. In the examples in this paper, this will always be written as “`<next>`”. Writing “`<next>`” on its own refers to the operator $\bigcirc$.

5 Two Simple scenarios

A simple rule of expectation that might apply in a Second Life scenario is that no one should ever fly. This might apply in a region used by members of a group that enacts historical behaviour. To monitor this expectation we can use the following property:

```
ExistsViol<next>(true, !any_flying)
```

This is an unconditional rule (it is triggered in every state) stating the expectation that there will not be any member of the group "Any" (comprising all avatars) flying.

If this is the only animation state to be tracked, the script's filter list will state that the animation "Flying" for group "Any" should be recorded, but otherwise all animations for all avatars and other groups should be discarded. On startup, the script sends the property to be monitored to the monitor, via Twitter, and then as avatars move around in Second Life and their animations are detected, it sends state messages that will either contain no propositions (if no one is flying) or will state that someone is flying:

```
S: any_flying
```

These states are accumulated, and each time a new state is received, the monitor is called and provided with the property to be monitored and the model (state history), e.g. s_1 : {}, s_2 : {}, s_3 : {`any_flying`} (the model is actually represented in XML—an example appears below).

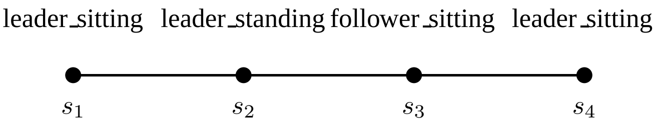
For this model, the monitor detects that the property is satisfied (i.e. the rule is violated) in state s_3 and a notification is sent back to the script. How this is handled is up to the script designer, but one option is for the script to be running in a "head-up-display" object, allowing the user to be informed in a way that other avatars cannot observe.

We now consider a slightly more complex example where there are two groups (or roles) specified in the script's group configuration list: `leader` (a singleton group) and `follower`. We want to monitor for violations of the rule that once the leader is standing, then from the next state a follower must not be sitting until the leader is sitting again. This is expressed using the following property:

```
ExistsViol<next>(
  leader_standing,
  <next>(U<next>(!follower_sitting,
                 leader_sitting))
)
```

The filter list can be configured so that only the propositions occurring in this rule are regarded as relevant for describing the state.

Suppose the scenario begins with the leader sitting and then standing, followed by the follower sitting, and finally the leader sitting again. This causes the following four states to be generated:



This is represented in the following XML format to be input to the model checker:

```
<hl-kripke-struct name="M">
  <world label="s1"/>
  <world label="s2"/>
  <world label="s3"/>
  <world label="s4"/>
  <modality label="next">
    <acc-pair to-world-label="s2"
              from-world-label="s1"/>
    <acc-pair to-world-label="s3"
              from-world-label="s2"/>
    <acc-pair to-world-label="s4"
              from-world-label="s3"/>
  </modality>
  <prop-sym label="leader_standing"
            truth-assignments="s2"/>
  <prop-sym label="leader_sitting"
            truth-assignments="s1 s4"/>
  <prop-sym label="follower_sitting"
            truth-assignments="s3"/>
  <nominal label="s1" truth-assignment="s1"/>
  <nominal label="s2" truth-assignment="s2"/>
  <nominal label="s3" truth-assignment="s3"/>
  <nominal label="s4" truth-assignment="s4"/>
</hl-kripke-struct>
```

The output of the model checker is:

```
  s3: (s2, U<next>(!(follower_sitting),
                   leader_sitting))
```

This means that a violation occurred in state s_3 from the rule being triggered in state s_2. The violated expectation (after progression to state s_3) is:

```
U<next>(!(follower_sitting), leader_sitting)
```

This information is sent to the script.

6 Discussion

As mentioned in Section 2, our detection script currently only detects the animations of avatars within sensor range. This limits the scenarios that can be modelled to those based on (simulated) physical action. However, it is straightforward to add the ability to detect other LSL events, provided that they can be translated to a propositional representation. Thus we could detect that an avatar has sent a chat message, but we can't provide a propositional encoding that can express all possible chat message contents. However, the addition of new types of configuration list would allow additional flexibility. For example, regular expressions or other types of pattern could be defined along with a string that can be appended to an avatar or group name to generate a proposition meaning that that avatar (or a member of that group) sent a chat message matching the pattern.

A significant limitation of the Linden Scripting Language is that the events that a script can detect are focused on the scripted object's own interactions with the environment—there is no facility for observing interactions between other agents, except for what can be deduced from their animations and chat. For many scenarios, it would be desirable to detect these interactions, for example, passing a certain object or sending money from one avatar to another might be a significant event in a society. One way around this problem would be to add additional scripted objects to the environment and set up the social conventions that these objects must be used for certain purposes. For example, an object in the middle of a conference table might need to be touched in order to request the right to speak next. These objects would generate appropriate propositions and send them to the main script via a private link.

The logic used currently is based on a discrete model of time, which can cause problems in some scenarios. For example, in the leader/follower scenario, it would be reasonable to allow the follower some (short) amount of time to stand after the leader stands. However, the moment that if a follower stands and another does not stand within the granularity of the same sensor event, then that second follower will be deemed in violation. It would be useful to be able to model some aspects of real time. This could be done by moving to a real-time temporal logic (which would involve some theoretical work on extending the model checker), or by some pragmatic means such as allowing the configuration parameters to define a frequency for regular "tick" timer events.

7 Related work

There seems to be little prior work that has explored the use of social awareness technology from multi-agent systems or other fields to support human interaction on the Internet in general, and in virtual worlds in particular.

A few avatar rating and reputation systems have been developed [16] to replace Second Life's own ratings system, which was disestablished in 2007. These provide various mechanisms to allow users to share their personal opinions of avatars with others.

Closer to our own work, Bogdanovych et al. [17, 18] have linked the AMELI electronic institution middleware [19] with Second Life. However, their aim is not to provide support for human interactions within Second Life, but rather to provide a rich interface for users to participate in an e-institution mediated by AMELI (in which the other participants may be software agents). This is done by generating a 3D environment from the institution's specification, e.g. *scenes* in the e-institution become rooms and transitions between scenes become doors. As a user controls their avatar to perform actions in Second Life, this causes an associated agent linked to AMELI to send messages to other agents, as defined by an action/message mapping table. Moving the avatar between rooms causes the agent to make a transition between scenes, but doors in Second Life will only open when the agent is allowed to make the corresponding scene transition according the rules of the institution.

This approach could be used to design and instrument environments that support structured human-to-human interaction in Second Life, but the e-institution model of communication is highly stylised and likely to seem unnatural for human users. In our work we are aiming to provide generic social awareness tools for virtual world users while placing as few restrictions as possible on the forms of interaction that are compatible with those tools. However, as discussed in Section 6, the limitation of the sensing functions provided by virtual world scripting languages may mean that some types of scenario cannot be implemented without providing specific scripted coordination objects that users are required to use, or the use of chat messages containing precise specified phrases.

8 Conclusion

This paper has reported on a prototype application of a model checking tool for social expectation monitoring applied to monitoring social interactions in Second Life. The techniques used for monitoring events in Second Life and allowing communication between a Second Life script and the monitor have been described, and these have been successfully tested on some simple scenarios. A discussion was presented on some of the limitation imposed by the LSL language and the logic used in the model checker, along with some suggestions for resolving these issues.

References

1. Linden Lab: Second Life home page. `http://secondlife.com/` (2008)
2. Boella, G., van der Torre, L., Verhagen, H.: Introduction to normative multiagent systems. In Boella, G., van der Torre, L., Verhagen, H., eds.: Normative Multi-agent Systems. Number 07122 in Dagstuhl Seminar Proceedings, Internationales Begegnungs- und Forschungszentrum für Informatik (IBFI), Schloss Dagstuhl, Germany (2007)
3. Cranefield, S., Winikoff, M.: Verifying social expectations by model checking truncated paths. In: Coordination, Organizations, Institutions, and Norms in Agent Systems IV. Volume 5428 of Lecture Notes in Computer Science. Springer (2009) 204–219
4. Linden Lab: LSL portal. `http://wiki.secondlife.com/wiki/LSL_Portal` (2008)
5. Twitter: Twitter home page. `http://twitter.com/` (2008)
6. Amazon Web Services: Amazon simple queue service. `http://aws.amazon.com/sqs/` (2008)
7. Yamamoto, Y.: Twitter4j. `http://yusuke.homeip.net/twitter4j/en/` (2008)
8. Alberti, M., Chesani, F., Gavanelli, M., Lamma, E., Mello, P., Torroni, P.: Compliance verification of agent interaction: a logic-based software tool. In Trappl, R., ed.: Cybernetics and Systems 2004. Volume II., Austrian Society for Cybernetics Studies (2004) 570–575
9. Esteva, M., de la Cruz, D., Sierra, C.: ISLANDER: an electronic institutions editor. In: Proceedings of the 1st International Joint Conference on Autonomous Agents and Multiagent Systems, ACM (2002) 1045–1052
10. Vázquez-Salceda, J., Aldewereld, H., Dignum, F.: Implementing norms in multiagent systems. In: Proceedings of the Second German Conference on Multiagent System Technologies (MATES). Volume 3187 of Lecture Notes in Computer Science., Springer (2004) 313–327

11. Farrell, A.D.H., Sergot, M.J., Sallé, M., Bartolini, C.: Using the event calculus for tracking the normative state of contracts. International Journal of Cooperative Information Systems **14**(2 & 3) (2005) 99–129
12. Bacchus, F., Kabanza, F.: Using temporal logics to express search control knowledge for planning. Artificial Intelligence **116**(1-2) (2000) 123–191
13. Dragone, L.: Hybrid logics model checker. http://luigidragone.com/hlmc/ (2005)
14. Markey, N., Schnoebelen, P.: Model checking a path. In: CONCUR 2003 – Concurrency Theory. Volume 2761 of Lecture Notes in Computer Science. Springer (2003) 251–265
15. Blackburn, P., de Rijke, M., Venema, Y.: Modal Logic. Cambridge University Press (2001)
16. Second Life: Removal of ratings in beta. `http://blog.secondlife.com/2007/04/12/removal-of-ratings-in-beta/` (2007)
17. Bogdanovych, A., Berger, H., Sierra, C., Simoff, S.J.: Humans and agents in 3D electronic institutions. In: Proceedings of the 4rd International Joint Conference on Autonomous Agents and Multiagent Systems, ACM (2005) 1093–1094
18. Bogdanovych, A., Esteva, M., Simoff, S.J., Sierra, C., Berger, H.: A methodology for 3d electronic institutions. In: Proceedings of the 6th International Joint Conference on Autonomous Agents and Multiagent Systems, IFAAMAS (2007) 358–360
19. Esteva, M., Rosell, B., Rodrguez-Aguilar, J.A., Arcos, J.L.: AMELI: An agent-based middleware for electronic institutions. In: Proceedings of the 3rd International Joint Conference on Autonomous Agents and Multiagent Systems. Volume 1., IEEE Computer Society (2004) 236–243

Normal = Normative?

The Role of Intelligent Agents in Norm Innovation

Marco Campennì[1,2], Giulia Andrighetto[1], Federico Cecconi[1], and Rosaria Conte[1]

[1] LABSS - Istituto di Scienze e Tecnologie della Cognizione - CNR, via S. Martino della Battaglia 44, 00185 Rome, Italy (http://labss.istc.cnr.it)
[2] University of Modena and Reggio Emilia, Italy

Abstract. In this paper the results of several agent-based simulations, aiming to test the role of normative beliefs in the emergence and innovation of social norms, are presented and discussed. Rather than mere behavioral regularities, norms are here seen as behaviors spreading to the extent that and because the corresponding commands and beliefs do spread as well. On the grounds of such a view, the present work will endeavour to show that a sudden external constraint (e.g. a barrier preventing agents from moving among social settings) facilitates norm innovation: under such a condition, agents provided with a module for telling what a norm is can generate new (social) norms by forming new normative beliefs, irrespective of the most frequent actions.

1 Introduction

Traditionally, the scientific domain of normative agent systems presents two main directions of research. The first is focused on intelligent agent architectures, and in particular on normative agents and their capacity to decide on the grounds of norms and the associated incentive or sanction. The second is focused on much simpler agents and the emergence of regularities from agent societies.

Very often, social scientific study of norms goes back to the philosophical tradition that defines norms as regularities emerging from reciprocal expectations [22, 5, 15]. Indeed, interesting sociological works [23] point to norms as public goods, the provision of which is promoted by 2nd-order cooperation [18, 19]. This view has inspired the more recent work of evolutionary game-theorists [17], who explored the effect of *punishers* or *strong reciprocators* on the group's fitness, but did not account for the individual decision to follow a norm.

No apparent contamination and integration between these different directions of investigation has been achieved so far. In particular, it is unclear how something more than regularities can emerge in a population of intelligent autonomous agents and whether agents' mental capacities play any relevant role in the emergence or innovation of norms.

In this paper, we will concentrate on *one* of these capacities, norm recognition. We will simulate agents endowed with the capacity to tell what a norm is, while observing their social environment.

Dagstuhl Seminar Proceedings 09121
Normative Multi-Agent Systems
http://drops.dagstuhl.de/opus/volltexte/2009/1898

One might question why start with norm recognition. After all, isn't it more important to understand *why* agents observe norms? Probably, it is. However, whereas this question has benn answered to some extent [10, 9] the question how agents tell norms has received poor attention so far. Furthermore, the account for the reason why agents observe the norms sheds poor light on our problem: norms need to have emerged, before they are complied with for any reason.

In this paper, we will address the antecedent phenomenon, norm innovation, postponing the consequent, norm compliance, to future studies. In particular, we will endeavour to show the impact of norm recognition on norm innovation. More precisely, we will observe agents endowed with the capacity to recognize a norm (or a behavior based on a norm); generate by herself new normative beliefs and transmit them to other agents by communicative acts or direct behaviors.

We intend to show whether a society of such normative agents allows norms to emerge or innovate. (By norm innovation, we mean the process by means of which the (set of) norm(s) shared within a (sub-)population changes in all or in part at any given time). Hence, we intend to investigate not only how norms come into existence, but also how they are maintained or replaced by other norms. The notion of norms that we refer to [11] is rather general. Unlike a *moral* notion, which is based based on the sense of right or wrong, norms are here meant in the broadest sense, as behaviors spreading to the extent that and because (a) they are prescribed by one agent to another, (b) and the corresponding normative beliefs spread among these agents.

Again, one might ask why not to address our moral sense, our sense of the right or wrong. The reason is at least twofold. First, our norms are more general, including moral and social norms. Secondly, and moreover, agents can deal with norms even when they have no moral sense: they can even obede norms they believe to be injust. But in any case, they must know what a norm is.

2 Existent Approaches

Usually, in the formal social scientific field, that is in utility and (evolutionary) game theory [5, 15, 25, 26, 28], the spread of new norms and other cooperative behaviors is not explained in terms of internal representations. The object of inquiry is usually the conditions for agents to converge on given behaviors, which proved efficient in solving problems of coordination [22] or cooperation [4], independent of the agents normative beliefs and goals [6]. In this field, no theory of norms based on mental representations (of norms) has yet been provided.

Game theorists essentially aimed to investigate the dynamics involved in the problem of norm convergence. They consider norms as conditioned preferences, i.e. options for action preferred as long as they are believed to be preferred by others as well [5]. Here, the main role is played by sanctions: what distinguishes a norm from other cultural products like values or habits is the fact that adherence to a social norm is enforced by sanctions [16, 3] and the utility function, which an agent seeks to maximize, usually includes the cost of sanction as a crucial component.

In the field of multi-agent systems [14, 21, 27], instead, norms are explicitly represented. However, they are implemented as built-in mental objects. This alternative approach has been focused on the question as to how autonomous intelligent agents decide on the grounds of their explicitly represented norms. Even when norm emergence is addressed [24], the starting point is some preexisting norms, and emergence lies in integrating them. When agents (with different norms) coming from different societies interact with each other, their individual societal norms might change, merging in a way that might prove beneficial to the societies involved (and the norm convergence results in the improvement of the average performance of the societies under study). Lately, decision making in normative systems and the relation between desires and obligations has been studied within the BDI framework, developing an interesting variant of it, i.e. the so-called Belief-Obligations-Intentions-Desires or BOID architecture [7].

In none of these approaches, including the last one, it is possible for an agent to tell that a given input is a (new) norm. On the contrary, obligations are hardwired into the agents' minds when the system is off-line. Unlike the game-theoretic model, multi-agent systems certainly exhibit all of the advantages deriving from an explicit representation of norms. Nevertheless, they overshadow one of the advantages of autonomous agents, i.e. their capacity to filter external requests. Such a filtering capacity affects not only normative decisions, but also the acquisition of new norms. Indeed, agents take decisions even when they decide to form normative beliefs, and then new (normative) goals, and not only when they decide whether to execute the norm or not [12].

Despite the undeniable significance of the results achieved, these studies leave some fundamental questions still unanswered, such as how and where norms originate, how agents acquire norms, and more specifically, how agents tell that something is a norm. Our feeling is that the question how norms are created and innovated has not received so far the answer it deserves the role of norm-recognition has been insufficiently perceived.

3 Objectives

Some preliminary simulations, discussed in [1], compared the behavior of a population of normative agents provided with a norm recognition module and a population of social conformers whose behavior is determined only by a rule of imitation. The results of these simulations show that under specific conditions, i.e. moving from one social setting to another, imitators are not able to converge on one behavior, even if this is common to different settings, whereas normative agents are.

In this paper we want to find out the sufficient (even if not necessary) conditions for existing norms to change. In particular, we want to show if a simple cultural or material constraint can facilitate norm innovation. To see this, we imagined a simple case in which subpopulations are isolated in different contexts for a fixed period of time. The methaphor here is any physical catastrophe or political upheaval that divides one population into two separate communities. The recent European history has shown several examples of this phenomenon.

4 Norm Innovation

Norms are highly adaptable artifacts, emerging, evolving, and decaying. If it is relatively clear how legal norms are put into existence, it is much less obvious how the same process applies to social norms. How do new social norms and conventions come into existence? Some simulation studies about the selection of conventions have been carried out, for example Epstein and colleagues' study of the emergence of social norms [15], and Sen and Airiau's study of the emergence of a precedence rule in the traffic [25]. However, such studies investigate which one is chosen out of a set of alternative equilibriums. A rather different sort of question concerns the innovation of social norms when no alternative equilibriums are available for selection.

We propose that a possible answer might be discovered while examining the interplay of communicated and observed behaviors, and the way they are represented into the minds of the observers. If any new behavior α is interpreted as obeying a norm, a new normative belief will be generated and a process of normative influence will be activated [13]. Such a behavior will be more likely to be replicated than would be the case if no normative belief were formed [2]. As shown elsewhere [9, 2], when a normative believer replicates α, she will be likely to influence others to do the same not only by ostensibly exhibiting the behavior in question, but also by explicitly conveying a norm. People impose new norms on one another by means of deontics and explicit normative valuations and propose new norms (implicitly) by means of (normative) behaviors. Of course, having formed a normative belief is necessary but not sufficient for normative influence: we will not answer the question *why agents do* so (a problem that we solve for the moment in prbabilistic terms), but we address the question how they can influence others to obey norms. They can do so if they have formed the corresponding normative belief, if they know how one ought to behave.

5 Normative Architecture

We consider a norm as a social behavior that spreads trough a population thanks to the diffusion of a particular belief, i.e. the normative belief. A normative belief, in turn, is a belief that a given behavior, in a given context, for a given set of agents, is either forbidden, obligatory, permitted, etc. Thus, for a norm-based behavior to take place, a normative belief has to be generated into the minds of the norm addressees and the corresponding normative goal has to be formed and pursued. Our claim is that a norm emerges as a norm only when it is incorporated into the minds of the agents involved [10, 11]; in other words, when agents recognize it as such. In this sense, norm emergence and stabilization implies its *immergence* [8] into the agents' minds.

5.1 Norm Recognizer

Our normative architecture (EMIL-A) (see [2] for a detailed description) consists of mechanisms and mental representations allowing norms to affect the behaviors of autonomous intelligent agents. EMIL-A is meant to show that norms not only regulate the behavior but also act on different aspects of the mind: recognition, adoption, planning, and decision-making. Unlike BOID in which obligations are already implemented into the agents' minds, EMIL-A is provided with a component by means of which agents infer that a certain norm is in force even when it is not already stored in their normative memory. In this situation the norm has not already been incorporated into schemata, scripts, or other pragmatic structures [5]; hence, agents are not facilitated by any of these. Actually, the norm needs to be found out, and only thereafter, stored. To implement such a capacity is conditioned to modeling agents' ability to recognize an observed or communicated social input as normative, and consequently to form a new normative belief. In this paper, we will only describe the first component of EMIL-A, i.e. the norm recognition module. This is most frequently involved in answering the open question we have raised, i.e. how a new norm is found out and we claim that to answer this question is particularly crucial in norm emergence, innovation and stabilization.

Our Norm Recognizer (see Fig. 1) consists of three layers and a link to the normative board, which is part of the agents long term memory. The normative board contains normative beliefs and normative goals, ordered by *salience*. With salience we refer to the degree of activation of a norm: in any particular situation, one norm may be more frequent than others, its salience being higher. The difference in salience between normative beliefs and normative goals has the effect that some of these normative mental objects will be more active than others and they will interfere more frequently and with more strength with the general cognitive processes of the agent.

In the higher layer, actions (α) presented as deontics (D) or normative valuations (V) are stored; in the lower layer, instead, actions are stored only if they have already been stored at the higher level, i.e., if they have been received by the agent as deontics or normative valuations. We identify six possible modals: assertions (A), i.e. generic sentences pointing to or describing states of the world; behaviors (B), i.e. actions or reactions of an agent, with regard to another agent or to the environment; requests (R), i.e. requests of action made by another agent; deontics (D), partitioning situations between good/acceptable and bad/unacceptable (we further distinguish deontics into three types: obligations, forbearances, permissions); normative valuations (V), i.e. assertions about what it is right or wrong, correct or incorrect, appropriate or inappropriate (i.e. *it is correct to respect the queue*).

Aiming to decide which action to produce, the agent will search through the normative board: if more than one is found out, the most salient norm will be chosen. Once received the input, the agent will compute the information in order to generate/update her normative beliefs. Every time a message containing a deontic (D) or a normative valuation (V) is received, the relative action will be stored as a (possible) norm. This will sharpen agents' attention: further messages with the same content, especially when observed as open behaviors, will be processed and stored at the same level. Beyond a certain normative threshold (which represents the frequency of corresponding normative behaviors observed, e.g. n% of the population), they will generate a new normative belief.

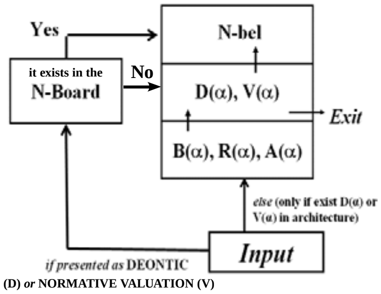


Fig. 1. The norm recognition module (in action): on the right side of the figure, from the bottom the *Input* and the two layers of the module (layer 0 and layer 1) plus the normative belief (generated or recognized); on the left side, the normative board. Vertical arrows in the block on the right side indicate the process regulating the generation of a new normative belief. The input action (α) can match with a norm present in the normative board (see the arrows path on the left side of the figure); or a new normative belief can be formed if the agent receives an input action (α) (at least one time as deontic or normative valuation) for a given number of times (as fixed by the normative threshold; see the arrows path on the right side of the figure). If the agent receives no other occurence of the same input action (α), after a fixed time t action α exits from the higher level and the process is finalized (see *Exit*).

6 The Model

In our simulation model, the environment consists of four scenarios, in which the agents can produce three different kinds of actions. We define two context-specific actions for every scenario, and one action common to all scenarios. Therefore, we have nine actions. Suppose that the first context is a postal office, the second an information desk, the third our private apartment, and so on. In the first context the action *stand in the queue* is a context-specific action, whereas in the second a specific action could be *occupy a correct place in front of the desk*. A common action for all of the contexts could be, *answer when asked*. Each of our agents is provided with a personal agenda (i.e. a sequence of contexts), an individual and constant time of permanence in each scenario (when the time of permanence is expired, the agent moves to the next context) and a window of observation (i.e. a capacity for observing and interacting with a fixed number of agents) of the actions produced by other agents. Norm Recognizers are also provided with the three-layer architecture described above, necessary to analyze the received information, and a normative board in which the normative beliefs, once arisen, are stored. The agents can move across scenarios: once expired the time of permanence in one scenario, each agent moves to the subsequent scenario following her agenda. Such irregular flow (each agent has a different time of permanence and a different agenda) generates a complex behavior of the system, tick-after-tick producing a fuzzy definition of the scenarios, and tick-for-tick a fuzzy behavioral dynamics.

We have modeled two different kinds of environmental conditions. In the first set of simulations, agents can move through contexts (following their personal agenda and in accordance with the personal time of permanence). In the second set of simulations, from a fixed time t, agents are obliged to remain in the context they have reached, till the end of the simulation: in this case agents can explore the contexts exchanging messages with one another and observing others' behaviors. When they reach the last context at time t, they can interact with same-context agents till the end of the simulation.

At each tick, the Norm Recognizers (NRs), paired randomly, interact exchanging messages. These inputs are represented on an ordered vector, consisting of four elements: the source (x); the modal through which the message is presented (M); the addressee (y); the action transmitted (a).

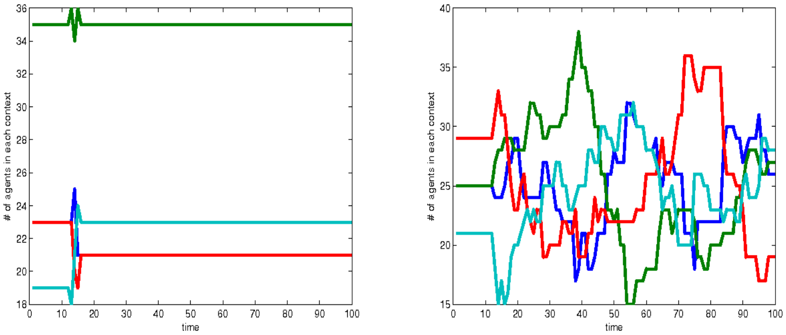
Codifying the input in such a way allows us to (a) access the information even later, if necessary; (b) recognize the source, a piece of information that might be useful to store inputs from recognized authorities; (c) account for a variety of information, thanks to the modals' syntax; (d) compute the received information in order to generate a new normative belief. NRs produce different behaviors: if the normative board of an agent is empty (i.e. it contains no norms), the agent produces an action randomly chosen from the set of possible actions (for the context in question); in this case, also the modal by means of which the action is presented is chosen randomly. Vice versa, if the normative board contains some norms, the agent chooses the action corresponding to the most salient among these norms. In this case the action produced is presented with one of these modals: deontic (D), normative valuation (Vn) or behavior (B). This corresponds to the intuition that if an agent has a normative belief, there is a high propensity (in this paper, this has been fixed to 90%) for her to transmit it to other agents under strong modals (D or Vn) or open behavior (B). We run several simulations for different values of the threshold, testing the behaviors of the agents in the two different experimental conditions.

7 Results and Discussion

We briefly summarize the simulation scheme. The process begins by producing actions (and modals) at random. The process is synchronic. The process is more and more complex runtime: agent i provides inputs to the agent who precedes her (k=*1*), issuing one action and one modal. Action choice is conditioned by the state of her normative board. When all of the agents have executed one simulation update, the whole process restarts at the next step.

7.1 Simulations' Results

Figure 2(a) and Figure 2(b) show the trend of simulation in terms of number of agents in each context runtime in both cases (the first with the external barrier, the second without it).



(a) with barrier

(b) without barrier

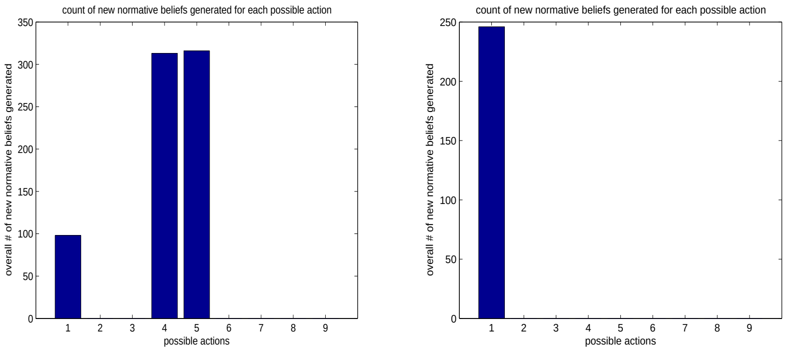
Fig. 2. Number of agents in each context runtime - with (left) and without (right) external barrier

First of all we present the results obtained when imposing the external barrier. Then, we present the results obtained when no barrier was imposed; finally we compare the former with the latter results.

Figure 3(a) shows the overall number of different new normative beliefs generated at the end of the simulation: as we can see, in the barrier condition, agents form more than one normative belief, whereas in the no barrier condition they form one normative belief only.

Figure 4 shows the trend of new normative beliefs generation runtime for a certain value of the norm threshold, which is a good implementation of our theory: each line represents the generation of new normative beliefs corresponding to an action (i.e. each line corresponds to the sum of different normative beliefs present in all of the agents). To be noted, a normative belief is not necessarily universally shared in the population. However, norms are behaviors that spread thanks to the spreading of the corresponding normative belief. Therefore, they imply shared normative beliefs.

Figures 7(a) and 7(b) are very similar (even if in the no-barrier variant, we find some noise in the chromatic definition of different contexts). In these figures, we cannot appreciate significant chromatic differences pointing to the normative



(a) with barrier

(b) without barrier

Fig. 3. Overall number of new normative beliefs generated for each type of possible action - with (left) and without (right) external barrier

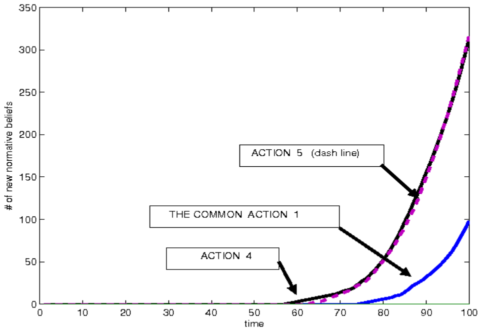
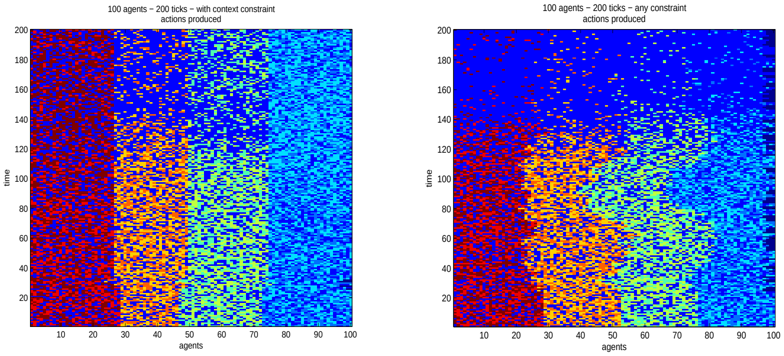


Fig. 4. New normative beliefs generated runtime - with external constraint



(a) with barrier

(b) without barrier

Fig. 5. Chromatic representation of the actions generated by the NRs. A different action corresponds to each color: the dark blue color represents the action common to the 4 scenarios; on axis X we find the number of agents (100) and on axis Y the number of simulation ticks (200) - with (left) and without (right) external barrier

beliefs acting on the effective behaviors: we cannot distinguish the chromatic effect corresponding to the agents' convergence on a specific norm. This is due to the length of these simulations, which is not sufficient to include the latency time of norms. In the previous study, indeed, we showed that for a normative belief to affect behavior, a certain number of ticks has to elapse, which we might call *norm latency*. Indeed, if we run longer simulations, we can appreciate the consequences of the results of our investigation: in Figures 5(a) and 5(b) we can observe two chromatic effect: (a) more or less at the same time both in the barrier (left) and no barrier (right) condition, a convergence on the common action (dark blue) is forming, much more homogeneous in (5.b) than in (5.a); (b) however, in the barrier condition, other areas of convergence are also emerging (e.g. a light blue in the last column).

This corresponds to what is shown in Figure 4 and Figure 6 on one hand, and Figure 3(a) and Figure 3(b) on the other: with external barrier, we can see that the higher overall number of new normative beliefs generated does not correspond to the common action (action 1) and the trend of new normative beliefs generated runtime shows the same results. With no external barrier, instead, only normative beliefs concerning action 1 are generated.

8 Concluding Remarks

We show that the model allows new norm, to emerge, despite another norm had previously emerged. More interestingly, the new norms do not correspond to the common action. Some rival norms now compete in the same social settings. Obviously, they will continue to compete, unless some further external event or

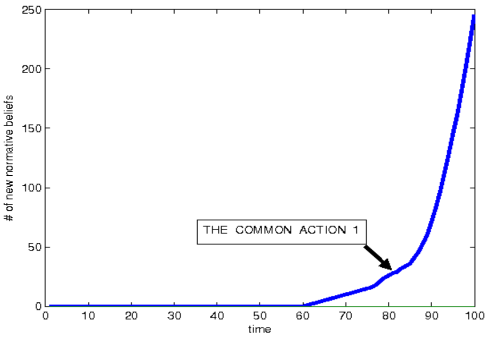
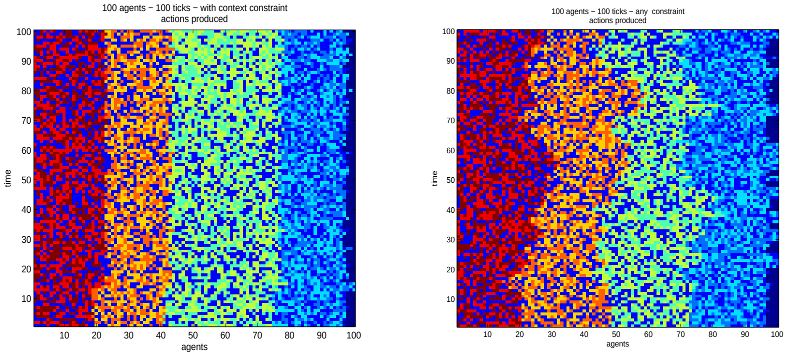


Fig. 6. New normative beliefs generated runtime - without external barrier



(a) with barrier

(b) without barrier

Fig. 7. Chromatic representation of the actions generated by the NRs. A different action corresponds to each color: the dark blue color represents the action common to the 4 scenarios; on axis X we find the number of agents (100) and on axis Y the number of simulation ticks (100) - with (left) and without (right) external barrier

change in the population (e.g. the barrier removal) will cause agents to start migrating again. It would be interesting to observe how long the rival norms will survive after barrier removal, whether and when one will out-compete the others, and if so, which one. It should be observed that, as we observe a latency time for a normative belief to give rise to a new normative behavior, we also expect some time to elapse before a given behavior disappears while and because the corresponding belief, decreasingly fed by observation and communication, starts to extinguish as well. We might call such a temporal discrepancy *inertia of the norm*. Both latency and inertia are determined by the twofold nature of the norm, mental and behavioral, which reinforce each other, thus preserving agents' autonomy: external barriers do modify agents' behaviors, but only through their minds.

More than emergence, our simulation shows a norm innovation process; in fact, Figure 4 shows that, starting around tick=60, two normative beliefs appear in the normative boards and the overall number of these two new normative beliefs generated is three times higher than the overall number of normative beliefs concerning the common action 1 (Figure 3(a)). Analogously, in Figure 5(a) some areas of homogeneity start to appear beyond the dark blue one.

We might say that, if stuck to their current location by external barriers, norm recognizers resist the effect of majority and do not converge on one equilibrium only. Rather, they will form as many normative beliefs as there were competing beliefs on the verge of overcoming the normative threshold before the agents had been stuck to their locations. No such effect is expected among agents whose behavior depends only from the observation of others.

In sum, is statistical frequency sufficient for a norm to emerge? Beside action 1, common to the four contexts, other norms seem to emerge in our simulation.

Hume seemed to doubt it [20].

Normative agents can recognize a norm; infer the existence of a norm by its occurrences in open behavior under certain conditions (see the critical role of previous deontics); and finally spread a normative belief to other agents.

Future studies are meant to investigate on the effect of barrier removal and the inertia of normative beliefs.

Acknowledgments

This work was supported by the EMIL project (IST-033841), funded by the Future and Emerging Technologies program of the European Commission, in the framework of the initiative Simulating Emergent Properties in Complex Systems.

References

1. G. Andrighetto, M. Campennì, F. Cecconi, and R. Conte. How agents find out norms: A simulation based model of norm innovation. In *3rd International Workshop on Normative Multiagent Systems (NorMAS 2008)*, submitted - 2008.

2. G. Andrighetto, M. Campennì, R. Conte, and M. Paolucci. On the immergence of norms: a normative agent architecture. In *Proceedings of AAAI Symposium, Social and Organizational Aspects of Intelligence Washington DC*, 2007.
3. R. Axelrod. An evolutionary approach to norms. *The American Political Science Review*, 4(80):1095–1111, 1986.
4. R. Axelrod. *The Evolution of Strategies in the Iterated Prisoner's Dilemma*. Kaufmann: Los Altos, CA, 1987.
5. C. Bicchieri. *The Grammar of Society: The Nature and Dynamics of Social Norms*. Cambridge University Press, New York, 2006.
6. K. Binmore. *Game-Theory and Social Contract. Vol. 1: Fair Playing*. Clarendon: Cambridge., 1994.
7. J. Broersen, M. Dastani, J. Hulstijn, Z. Huang, and L. van der Torre. The boid architecture. conflicts between beliefs, obligations, intentions and desires,. In *In Proceedings of the fifth international conference on Autonomous agents, Montreal, Quebec, Canada. .*, pages 9 – 16, 2001.
8. C. Castelfranchi. Simulating with cognitive agents: The importance of cognitive emergence. Multi-Agent Systems and Agent-Based Simulation, Heidelberg, 1998.
9. C. R. Castelfranchi, C. From conventions to prescriptions. towards a unified theory of norms. *AI and Law*, 7:323–340, 1999.
10. R. Conte and C. Castelfranchi. *Cognitive and social action*. University College of London Press, London, 1995.
11. R. Conte and C. Castelfranchi. The mental path of norms. *Ratio Juris*, 19(4):501–517, 2006.
12. R. Conte, C. Castelfranchi, and F. Dignum. Autonomous norm-acceptance. In *Proceedings of the 5th International Workshop on Intelligent Agents V, Agent Theories, Architectures, and Languages*, pages 99–112, 1998.
13. R. Conte and F. Dignum. From social monitoring to normative influence. *Jasss - the Journal of Artificial Societies and Social Simulation*, 4(2), 2001.
14. F. Dignum. Autonomous agents with norms. *Artificial Intelligence and Law*, 7(1):69–79, 1999.
15. J. Epstein. *Generative Social Science. Studies in Agent-Based Computational Modeling*. Princeton-New York: Princeton University Press., 2006.
16. T. Feld. Collective social dynamics and social norms. *Munich Oersonal RePEc Archive*, 2006.
17. B. S. B. R. F. E. Gintis, H. Explaining altruistic behavior in humans. *Evolution and Human Behavior*, (24):153–172, 2003.
18. D. Heckathorn. Collective sanctions and the compliance norms - a formal theory of group-mediated social-control. *American Journal of Sociology*, (94):535–562, 1988.
19. C. Horne. Explaining norm enforcement. *Rationality and Society*, (19(2)):139–170, 2007.
20. D. Hume. *A treatise of Human Nature*. Oxford University Press, [1740] 1978.
21. A. Jones and M. Sergot. A formal characterization of institutionalized power. *Logic Journal of the IGPL.*, 4(3):429–445, 1996.
22. D. K. Lewis. *Convention: A Philosophical Study*. Cambridge Mass.: Harvard University Press., 1969.
23. P. E. Oliver. Formal models of collective action. *Annual Review of Sociology*, (19):271–300, 1993.
24. B. Savarimuthu, M. Purvis, S. Cranefield, and M. Purvis. How do norms emerge in multi-agent societies? mechanisms design. *The Information Science Discussion Paper*, (1), 2007.

25. S. Sen and S. Airiau. Emergence of norms through social learning. In *Proceedings of the Twentieth International Joint Conference on Artificial Intelligence*, 2007.
26. E. Ullman-Margalit. *The Emergence of Norms.* Clarendon Press, Oxford, 1977.
27. L. Van der Torre and Y. Tan. Contrary-to-duty reasoning with preference-based dyadic obligations. *Annals of Mathematics and Artificial Intelligence*, pages 1239–1246, 1999.
28. H. P. Young. *Individual Strategy and Social Structure: An Evolutionary Theory of Institutions.* Princeton University Press, 1998.

NorMAS-RE: a Normative Multiagent Approach to Requirements Engineering

Serena Villata

Department of Computer Science, University of Turin, Italy

Abstract. In this paper we present a new model, called NorMAS-RE, for the requirements analysis of a system. NorMAS-RE is a new model based on the multiagent systems paradigm with the aim to support the requirements analysis phase of systems design. This model offers a structured approach to requirements analysis, based on conceptual models defined following a visual modeling language, called dependence networks. The main elements of this visual language are the agents with their goals, capabilities and facts, similarly to the TROPOS methodology [10]. The normative component is present both in the ontology and in the conceptual metamodel, associating agents to roles they play inside the systems and a set of goals, capabilities and facts proper of these roles. This improvement allows to define different types of dependence networks, called dynamic dependence networks and conditional dependence networks, representing the different phases of the requirements analysis of the system. This paper presents a requirements analysis model based on normative concepts such as obligation and institution. The NorMAS-RE model is a model of semiformal specification featured by an ontology, a meta-model, a graphical notation and a set of constraints. Our model, moreover, allows the definition of the notion of coalition for the different kinds of network. We present our model using the scenario of virtual organizations based on a Grid network.

1 Introduction

The diffusion of software applications in the fields of e-Science and e-Research underlines the necessity to develop open architectures, able to evolve and include new software components. In the late years, the process of design of these software systems became more complex. The definition of appropriate mechanisms of communication and coordination between software components and human users motivates the development of methods with the aim to support the designer for the whole development process of the software, from the requirements analysis to the implementation.

The answer to this problem comes from software engineering that provided numerous methods and methodologies allowing to treat more complex software systems. One of these methodologies is the TROPOS methodology [10], developed for agent-oriented design of software systems. The intuition of the TROPOS methodology [10] is to couple, together with the instruments offered by software engineering, the multiagent paradigm. In this paradigm, the entities composing the system are agent, autonomous by definition, characterized by their own sets of goals, capabilities and beliefs. The multiagent paradigm allows the cooperation among the agents with the aim to obtain common

Dagstuhl Seminar Proceedings 09121
Normative Multi-Agent Systems
http://drops.dagstuhl.de/opus/volltexte/2009/1896

and personal goals. In this way, multiagent systems offer a solution for open, distributed and complex systems and the approach combining software engineering and multiagent systems is defined Agent-Oriented Software Engineering (AOSE). TROPOS [10] covers five phases of the software development process: the early requirements allowing to analyze and model the requirements of the context in which the software system will be inserted, late requirements describing the requirements of the software system, architectural design and detailed design aiming to design the architecture of the system and, finally, the code implementation.

The TROPOS methodology [10] is based on the multiagent paradigm consisting in a set of agents and their features but it does not consider the addition of a normative perspective to this paradigm. Since twenty years, the design of artificial social systems is using mechanisms like social laws and norms to control the behavior of multiagent systems [5]. These social concepts are used in the conceptual modeling of multiagent systems, for example in requirement analysis, as well as in formal analysis and agent based social simulation. For example, in the game theoretic approach of Shoham and Tennenholtz [28], social laws are constraints on sets of strategies. Together with the rationality assumptions of classical game theory, this leads to the analysis of, for example, stable or minimal social laws, which can be used to choose the best alternative among a set of available social laws. More recently, institutions have emerged as a new mechanism in the design of artificial social systems, which are used in conceptual modeling of multiagent organizations in agent oriented software engineering [37]. Roughly speaking, institutions are structures and mechanisms of social order and cooperation governing the behavior of a set of individuals. They are needed to enforce the global behaviour of the society and to assure that the global goals of the society are met. However, the formal analysis of the institutions is challenging due to the complexity of its dynamics. For example, the agents may change the roles they are playing, or the institution itself may change over time due to the behavior of the agents. Requirements analysis represents the initial phase in many software engineering methodologies. As with the other approaches, the ultimate objective of requirements analysis is to provide a set of functional and non-functional requirements for the system to be. In this paper, we propose to add institutions and norms, presented thanks to the normative multiagent paradigm, to the requirements analysis phase. This paper addresses the following research question:

- How to develop a new model for requirements analysis based on the normative multiagent paradigm?

Our approach is based, following the approach of TROPOS [10], on a semiformal language of visual modeling and it is composed by the following components. First, we present an ontology that defines the set of concepts used in the modeling. The elements composing the ontology are agents, goals, facts, skills, dependencies, coalitions with the addition of the normative notions of roles, institutional goals, institutional facts, institutional skills, dynamic dependencies, obligations, sanctions, secondary obligations and conditional dependencies. Second, our meta-model is specified by a number of UML diagrams. These diagrams and the graphical notation establish how to graphically depict the elements composing models. A NorMAS-RE model is a directed labeled graph whose nodes are instances of the metaclasses of the metamodel, e.g., agents, goals, facts, and whose arcs are instances of the metaclasses representing relationships between them such as dependency, dynamic dependency, conditional dependency. Finally, we have a set of rules and constraints to guide the building of a conceptual metamodel. In TROPOS [10], the requirements analysis is split in two main phases, the early requirements and the late requirements. In our model, these two phases share the same conceptual and methodological approach, thus we call both of them only requirements analysis.

We provide the abstract notion of institution and a definition of a new modeling, called dynamic dependency modeling, based on the structure of dynamic dependence networks. These networks, as classical dependence networks, depict the dependencies among the agents. The dependencies reflect the relation between the goals of agents and agents who have the power to achieve them. In the institutional perspective, institutional powers cannot be captured by the existing dependence networks formalism, since they introduce a dynamic component. Institutional powers can change the norms and permissions of agents playing roles, and, thus, by exercising a power an agent transforms a dependence structure into a new one by adding or removing dependencies thanks to the concepts present into the institutional level of the ontology. Thus, power is seen as the base of the change that is applied to the network describing the system, differently from what expresses by Jones and Sergot [20] and Grossi [19]. By exercising an institutional power, an agent transforms a dependence structure into a new one by adding or removing dependencies associated to the institutional concepts. Moreover, we introduce the normative issue of obligations, representing them directly in dependence networks. This introduction allows the definition of a third kind of modeling called conditional dependency modeling based on the structure of conditional dependence networks. Conditional dependence networks represent obligations as particular kind of dependencies and these obligations are related to notions as sanctions, if the obligation is not fulfilled, and as contrary to duty when the primary obligation, not fulfilled, actives a secondary obligation.

A coalition is an alliance among agents, during which they cooperate in joint action, each one following his own self-interest. We define the notion of coalition in dependence networks, based on the idea that to be part of a coalition, every agent has to contribute something, and has to get something out of it. Since the processes involving coalitions dynamics are complex and costly social behaviors, the idea is that agents have to maintain the stability of their own coalition, paying attention to the possible actions that can be performed by the other agents to strategically increase their profit, mining the coalition or, even worse, destroying the coalition itself. To maintain stability, coalitions have to change dynamically. The possibility to represent coalitions is relevant for systems design and, in particular, for the requirements analysis where the different components of the system can have the necessity to cooperate in a preferential way with a specific subset of other components. The aim of requirements analysis in this context consists in the definition of models able to represent these groups and to provide methods to maintain the stability and the cohesion of these groups. The introduction of methods of social order such as obligations and sanctions represents an efficient way to achieve this purpose.

Our model is not intended to support all analysis and design activities in software development process, from application domain analysis down to the system implementation as in the TROPOS methodology [10]. Moreover, we do not perform any kind of simulation as in the recent developments of social network analysis called dynamic networks analysis as in Carley [12]. Finally, the treatment of a topic like contrary to duty does not concern any connection with deontic logic approaches to solve and analyze this structure such as in Prakken and Sergot [23].

This paper is organized as follows. Section 2 describes a Grid computing scenario as case study for the design of virtual organizations for e-Science and e-Research. In Section 3, we present the core concepts of the ontology and their inter-relations. In Section 4, we define the structure of dynamic dependence networks and we introduce the notion of coalition in this kind of network. Section 5 presents a new kind of dependence network, called conditional dependence network, introducing some constraints that have to be set for representing coalitions in the conditional dependency modeling. A notion of coalitions' stability is defined and a discussion on the this issue is presented. Related work and conclusions end the paper.

2 The Grid Scenario

Grids and the Grid Computing paradigm provide the technological infrastructure to facilitate e-Science and e-Research. Grid technologies can support a wide range of research including amongst others: seamless access to a range of computational resources, linkage of a wide range of data resources, exploitation of shared instruments such as astronomical telescopes or specialized resources such as visualization servers. Historically, much of the focus and effort of Grid computing was based upon addressing access to and usage of large scale high performance computing (HPC) resources such as cluster computers. These access models are typified by their predominantly authentication-only based approaches which support secure access to an account on a cluster. It is often the case that research domains and resource providers require more information than simply the identity of the individual in order to grant access to use their resources. The same individual can be in multiple collaborative projects each of which is based upon a common shared infrastructure. Knowing in what context a user is requesting access to a particular resource is essential information for a resource provider to decide whether the access request should be granted or not. This information is typically established through the concept of a virtual organization (VO) [32]. A virtual organization allows the users, their roles and the resources they can access in a collaborative project to be defined.

In the context of virtual organizations, there are numerous technologies and standards that have been put forward for defining and enforcing authorization policies for access to and usage of virtual organizations resources. Role based access control (RBAC) is one of the more well established models for describing such policies. In the RBAC model, virtual organization specific roles are assigned to individuals as part of their membership of a particular virtual organization. Possession of a particular role, combined with other context information, such as time of day and amount of resource being requested, can then be used by a resource gatekeeper to decide whether an ac-

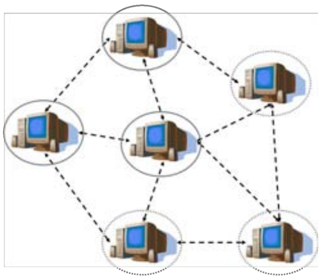


Fig. 1. A Grid composed by six nodes and the interconnections among them.

cess request is allowed or not. One of the key advantages is that whilst individuals in a virtual organization may come and go, the role itself is unlikely to change as much. Consequently RBAC based approaches are considered more scalable and manageable. The key advantage of RBAC-based security models compared to other approaches is that privileges and access is determined by roles and memberships a user holds and not merely on identity. Indeed the common philosophy underlying the Grid is that all resource providers are expected to be autonomous, i.e. they may allow/deny access requests at their own discretion. Nevertheless, a crucial consideration in establishing a virtual organization is whether a common understanding of the various roles and their associated privileges needs to be established throughout the entire virtual organization or not.

There are two primary models for defining roles specific to a virtual organization: the centralized and decentralized models [32]. In the centralized model, all sites agree in advance on the definition and names of the roles that are applicable to their particular virtual organization, and the privileges that will be assigned to them. A single virtual organization administrator is then appointed who will typically assign these roles to individuals on a case by case basis when users ask to be granted particular roles or permissions in the virtual organization. The decentralized virtual organization role model is more aligned with the original dynamic collaborative nature of the Grid. In this model there is no central virtual organization administrator. Instead, each resource site has its own local administrator who is completely responsible for determining which virtual organization members can access the local virtual organization resources. Each site administrator determines the roles and the associated privileges that are required to access and use the local resources. Consequently, they can decide which other administrators (at this and other virtual organization sites) are trusted to assign which roles to which virtual organization users. In this way they may each delegate to each other the responsibility of user-role assignments throughout the virtual organization. This model allows for more dynamic collaborations to occur. Thus rather than all sites having to agree on virtual organization-wide roles and develop associated policies, the decentralized model allows a resource administrator to directly provide end users and trusted end user administrators with the privileges they need to enable access to his resource.

Role based access control systems make access control decisions based on the roles that users hold. Traditional output of the access control decisions are *Granted* and *De-*

nied, which dictate whether the requests are authorized or not. As presented by Zhao et al. [38], obligations are requirements and tasks to be fulfilled, which can be augmented into conventional systems to allow extras information to be specified when responding to authorization requests. For example in [38], administrators can associate obligations with permissions, and require the fulfillment of the obligations when the permissions are exercised. The base model associated users with roles, and roles with permissions. Users, being members of roles, acquired all permissions associated with the roles. The hierarchical model enhanced the base model by allowing senior roles to acquire permissions of their junior roles. The general idea of the role based access control model is that, permissions are associated with functional roles in organizations, and members of the roles acquire all permissions associated with the roles. Allocation of permission to users is achieved by assigning roles to users. An obligation is associated with privileges, and when an operation is performed, the obligation associated to the privilege which authorizes the operation is activated. Obligations are requirements to be performed by a specific deadline. Failure of the fulfilling an obligation will incur a sanction.

Some of the main features of a node in a Grid are reliability, degree of accepted requests, computational capabilities, degree of faults and degree of trust for confidential data. These different features set up important differences among the nodes and the possible kinds of coalitions that can be formed and maintained. In this scenario, as in the following examples, we do not consider the way the coalitions are formed but we are interested in coalitions' evolution. We think of already formed coalitions and we discuss the notion of stability and the possible ways to regulate these coalitions thanks to the use of obligations. The idea is that coalitions emerge thanks to the preferred relationships among the different nodes, e.g., each node maintains a sort of list of the more trusted nodes forming a coalition with it. Reciprocity-based coalitions can be viewed as a sort of virtual organizations in which there is the constraint that each node has to contribute something, and has to get something out of it. For example, in a virtual organization each node has to be useful to the other and thus it has to have at least one of the previous cited features.

The scenario of virtual organizations based on Grid networks represents a case study able to underline the benefits of a normative multiagent paradigm for requirements analysis. First of all, in the normative multiagent paradigm as well as in the common multiagent one, the autonomy of agents is the fix point of all representations, i.e., the Grid philosophy imposes the autonomy of the nodes composing it. Second, the normative multiagent paradigm allows a clear definition of the notion or role and its associated permissions, i.e. the role based access control policy needs a design able to assign roles and represents all the consequent constraints based on them. Third, the normative multiagent paradigm allows the introduction at requirements analysis level of obligations able to model the system. Fourth, the concept of coalition and the constraints introduced by this concept to the early and late requirements model can design the concept of “local network" in virtual organizations. Finally, the modeling activities of dependency modeling, dynamic dependency modeling and conditional dependency modeling depict the system using structures similar to the Grid network itself.

3 Institutional MAS: agents, roles and assignments

Since last years, many factors have caused a great increase of the complexity of software systems. Applications such as e-commerce, e-services, e-science, e-research are clear example of this kind. The software for these applications has to be based on open architectures and it has to evolve over time to integrate new hardware components and answer to the necessity of new requirements. Our model is addressed to the representation of the requirements of the system using the normative multiagent paradigm. This model is based firstly on an ontology containing a number of concepts related to each other. We divide our ontology in three submodels: the agent model, the institutional model, and the role assignment model, as shown in Figure 2. The Figure depicts the three submodels which group the concepts of our ontology.

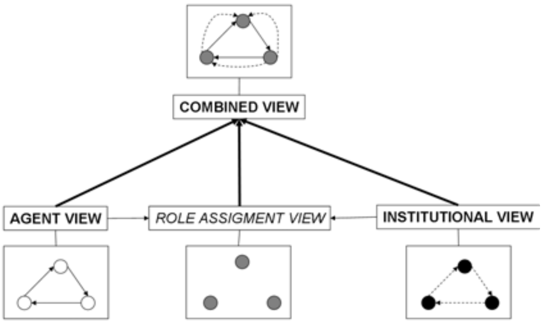


Fig. 2. The NorMAS-RE conceptual metamodel.

Such a decomposition is common in organizational theory, because the organization can be designed without having to take into account the agents that will play a role in it. Also, if another agent starts to play a role, for example if a node with the role of simple user becomes a VO administrator, then this remains transparent for the organizational model. Likewise, agents can be developed without knowing in advance in which institution they will play a role.

The notion of agent and all its features as goals, capabilities, are used in the conceptual modeling as in TROPOS [10]. In our model, we add to these notions those related to the institution such as the notion of role and all its institutional goals, capabilities and facts. Both these notions, combined in the combined view, are used in the conceptual modeling and to each agents it is possible to assign different roles depending on the organization in which the agent is playing. Adding the institution, to each agent are associated both a number of physical features and a role with all its institutional features. In this way, early and late requirements can be based both on agents and on roles. The models in NorMAS-RE are acquired as instances of a conceptual metamodel resting on the concepts presented in the following subsections. We present our three submodels as definitions and each definition contains the concepts belonging to this particular subset of the ontology.

3.1 Agent View

An agent can be defined as an entity characterized by a number of features as his capabilities, called skills, his world description and his goals, such as the tasks he want to achieve. The representation of the system from a material point of view, called Agent view, can be imagined as composed by a set of agents, each of them with its associated sets of skills and goals and a set of actions, a set of facts describing the world and a set of rules that allow the application of an action by an agent that can perform it and the consequences of the action on the system. The definition of the agent view is as follows:

Definition 1 (Agent view).
$\langle A, F, G, X,$ goals $: A \rightarrow 2^G,$ skills $: A \rightarrow 2^X, rules : 2^X \rightarrow 2^G\rangle$ consists of a set of agents A, a set of facts F, a set of goals G, a set of actions X, a function $goals$ that relates with each agent the set of goals it is interested in, a function $skills$ that describes the actions each agent can perform, and a set of rules, represented by the function $rules$ that relate sets of actions with the sets of goals they see to.

Example 1. Considering a virtual organization on a Grid with a role based access control policy, the agent view is used to describe the set of legitimate users of the system, represented inside the Grid as nodes. Each user is provided by a set of actions he can do, represented by the set X, e.g., to save a file on his file system or to start a computation on his personal computer, and by a set of goals he would fulfill, represented as the set G, e.g., he wants to reserve half of his available memory for his data or he has to obtain the result of a computation in two hours. These actions X can be compared to the operations that are recognized by the system. Functions $goals$ and $skills$ link each agent with the actions he can perform and with the goals he would obtain. Function $rules$ is a sort of action-consequence function, relating sets of actions with the goals they allow to fulfill, e.g., to obtain the results of a computation in two hours, the user has to start the computation on his personal computer.

3.2 Institutional View

A social structure is modeled as a collection of agents, playing roles regulated by norms where “interactions are clearly identified and localized in the definition of the role itself" [37]. The notion of role is notable in many fields of Artificial Intelligence and, particularly, in multiagent systems where the role is viewed as an instance to be adjoined to the entities which play the role. According to Ferber [17], “A role describes the constraints (obligations, requirements, skills) that an agent will have to satisfy a role, the benefits (abilities, authorizations, profits) that an agent will receive in playing that role, and the responsibilities associated to that role". In TROPOS [10], the role is one of the three specification of the concept of actor and it is an abstract characterization of the behaviour of the social actor inside the specific context of the application domain. In the NorMAS-RE model the notion of role is inserted into the submodel called institutional view. The institutional view is defined as follows:

Definition 2 (Institutional view).
$\langle RL, IF, RG, X, igoals : RL \rightarrow 2^{RG}, iskills : RL \rightarrow 2^X, irules : 2^X \rightarrow 2^{IF}\rangle$

consists of a set of role instances RL, a set of institutional facts IF, a set of public goals attributed to roles RG, a set of actions X, a function $igoals$ that relates with each role the set of public goals it is committed to, a function $iskills$ that describes the actions each role can perform, and a set of institutional rules $irules$ that relates a set of actions and the set of institutional facts they see to.

Example 2. The institutional view represents in the Grid scenario a sort of model for the role based access control policy. In fact, this view represents all the possible roles that can be instantiated in the system and all the possible actions and goals related to each of these roles. For example, we can think to a Grid system with the two basic roles of virtual organization administrator and virtual organization member. These two roles are different depending on the actions they can perform. For example, the VO administrator has the possibility to assign to the VO members the privileges they need to enable access to its resource. Our approach gives the opportunity to define not only the capabilities of a particular role but it allows also the definition of institutional goals associated to roles, differently from other approaches such as [38] [32]. The institutional view is a way to represent permissions of the users of the system. Users, being assigned to a particular role, acquire all permissions (in this view represented as rules by the function $irules$) associated to the role. In this way, the allocation of permissions to users is achieved by assigning roles to users. In the Grid computing field, a permission is an approval of performing an operation on a specific target. In our model, we represent a permission in a virtual organization as the actions that a role can perform and what kind of goals these actions allow to achieve. For example, a user asks for saving a file on the file system of another node. This user is associated to a role, since he belongs to a virtual organization regulated by a role based access control policy. The request can be processed either by the local VO administrator or by the user that has received the request. If the user requesting the service has a role that can perform this action, the request is accepted and the file is saved. In this case, we consider for simplicity the case in which the request is always accepted if the role has the permission to do it without thinking of malicious behaviours.

3.3 Role Assignment View

In TROPOS [10], the position of the actor represents a set of roles played by a single agent. In our model, we introduce the third submodel,the Role assignment view, which links the agent and the institutional view to each other, by relating agents to roles.

Definition 3 (Assignment view).
$\langle A, RL, roles\colon RL \rightarrow A\rangle$ *consists of a set of agents A, a set of role instances RL, and a function $roles$ assigning a role to its player in A.*

Example 3. The assignment view relates each agent with the role it is associated with. In virtual organizations, this kind of assignments is done by the VO administrator, in the centralize model, and by the VO local administrators, in the decentralized model. In our model, there is not a constraint on what kind of agent has the power to assign roles and thus privileges to the users. The assignment view can be eventually restricted to one of the two cases of centralized and decentralized model.

3.4 Combined View

In NorMAS-RE, the system is provided with two distinct views, the material one, called the agent view, and the institutional one, called institutional view, that aims to regulate the behaviour of the agents and to presents the permissions associated to each role. Usually, in a multiagent system each agent is related to a set of facts and goals the other agents cannot change since all the agents are autonomous. All these features are presented in the concepts of the agent view. But a multiagent system is composed by a multitude of agent that, thanks to their existence inside a social structure, are provided by new sets of facts and goals, the institutional ones, representing permissions. Permissions are allocated to roles and it is specified by the institutional view. The combined view unifies the agent view and the institutional view thanks to the assignment view providing thus the combined and unified conceptual metamodel:

Definition 4 (Combined view).
Let $\langle A, RL, roles : RL \rightarrow A\rangle$ be a role assignment view for the agents and role instances defined in the agent view $\langle A, F, G, X, \textit{goals}\colon A \rightarrow 2^G, \textit{skills}\colon A \rightarrow 2^X, rules : 2^X \rightarrow 2^G\rangle$ and institutional view $\langle RL, IF, RG, X, igoals : RL \rightarrow 2^{RG}, iskills : RL \rightarrow 2^X, irules : 2^X \rightarrow 2^{IF}\rangle$. The role playing agents are $RPA = \{\langle a, r\rangle \in A \times RL \mid r \in roles(a))\}$. The combined view associates with the role playing agents the elements of the agent and institutional view.

Example 4. The agents start with their sets of personal beliefs and goals and, only after their insertion inside a social structure, they enlarge their sets of goals and beliefs. In particular, the set of goals is enlarged with new normative goals that represent the responsibilities of the agent inside its social structure while the set of beliefs is enlarged with new normative beliefs representing the set of constitutive norms of the systems, norms based on the collective acceptance of the society representable by means of an institutional ontology.

3.5 Dependency Modeling

A NorMAS-RE model is a directed labeled graph whose nodes are instances of the metaclasses of the metamodel, e.g., agents, goals, facts, and whose arcs are instances of the metaclasses representing relationships between them such as dependency, dynamic dependency, conditional dependency. The building of a model in NorMAS-RE involves many activities contributing to the process of definition of the model itself. Our modeling is based on the theory of the social power and dependence pioneered by Castelfranchi [14] as starting point and then developed in the context of coalition formation by Sichman [29] and Sauro [25]. The theory of social power and dependence is an attempt to transfer theories developed initially in the field of sociology to the field of multiagent systems and to refine them. This theory models the potential interactions among the agents which lead to the achievement of a shared goal, i.e. cooperation, or the reciprocal satisfaction of their own goals, i.e. social exchange. This involves the development of a social reasoning mechanism that analyzes the possibility to profit from mutual-dependencies, e.g., the case in which two agents depend on each other for the satisfaction of a shared goal, or reciprocal-dependencies, e.g., the case in which two agents depend on each other for the satisfaction of two different goals.

In a multiagent system, since an agent is put into a system that involves also other agents, he can be supported by the others to achieve his own goals if he is not able to do them alone. This leads to the concept of power representing the capability of a group of agents (possibly composed only by one agent) to achieve some goals (theirs or of other agents) performing some actions without the possibility to be obstructed. The power of a group of agents is defined as follows:

Definition 5 (Agents' power).
$\langle A, G, power : 2^A \to 2^{2^G} \rangle$ where A is a set of agents, G is a set of goals. The function $power$ relates with each set $S \subseteq A$ of agents the sets of goals $G_S^1, \ldots, G_S^m$ they can achieve.

Example 5. In the Grid scenario, the simplest kind of example of power consists in the power of the local or global administrator to give to common users the possibility to access to a resource. Particularly, if we consider a role based access control policy, the Grid administrator has the power to give to the common users, under request, a new role which makes him able to access to a resource. Other kinds of powers are, for example, the power to perform a heavy computation or to memorize a great amount of data.

The notion of power brings to the definition of a structure with the aim to show the dependencies among agents. In order to define these relations in terms of goals and powers, we adopt, as said, the methodology of dependence networks as developed by Conte and Sichman [30]. In this model, an agent is described by a set of prioritized goals, and there is a global dependence relation that explicates how an agent depends on other agents for fulfilling its goals. For example, *dep*$(\{a, b\}, \{c, d\}) = \{\{g_1, g_2\}, \{g_3\}\}$ expresses that the set of agents $\{a, b\}$ depends on the set of agents $\{c, d\}$ to see to their goals $\{g_1, g_2\}$ or $\{g_3\}$. For each agent we add a priority order on its goals, and we say that agent a gives higher priority to goal g_1 than to goal g_2, written as $\{g_1\} \succ (a) \ \{g_2\}$, if the agent tries to achieve goal g_1 before it tries to achieve g_2. In other words, it gives more attention to g_1 than to g_2. A dependence network is defined as follows:

Definition 6 (Dependence Networks (DN)).
A dependence network is a tuple $\langle A, G, dep, \geq \rangle$ where:

- *A is a set of agents;*
- *G is a set of goals;*
- *dep : $2^A \times 2^A \to 2^{2^G}$ is a function that relates with each pair of sets of agents all the sets of goals on which the first depends on the second.*
- *$\geq$: $A \to 2^G \times 2^G$ is for each agent a total pre-order on goals which occur in his dependencies: $G_1 \geq (a) G_2$ implies that $\exists B, C \subseteq A$ such that $a \in B$ and $G_1, G_2 \in depend(B, C)$.*

Dependence networks represent our first modeling activity, the *dependency modeling*, consisting in the identification of the dependencies among the agents and among the roles. In the early requirements phase, we model the dependencies among the agents and the roles associated to the agents of the organization. In this way, we represent the domain stakeholders and we model them using the multiagent paradigm with the addition of the normative component with its related concepts. These dependencies are based both on goals and institutional goals. In the phase of late requirements, the same kind of approach is followed but the agents involved in the dependence network are those of the future system. A graphical representation of the model obtained following the *dependency modeling* is built following the legend of Figure 3 which describes the agents (depicted as white circles), the roles (depicted as black circles), the agents assigned to roles (depicted as grey circles), the agents'/roles' goals (depicted as white rectangles) and the dependency among agents (one arrowed line connecting two agents with the addition of a label which represents the goal on which there is the dependency). For simplicity, the legend considers the dependency only among agents but these dependencies can be also among roles or agents assigned to roles.

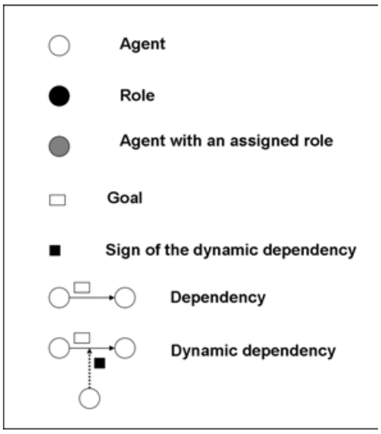


Fig. 3. The legend of the graphical representation of the modeling activities of *dependency* and *dynamic dependency*.

We present a first example of modeling a virtual organization based on a Grid network containing only the notions of the agent view.

Example 6. Considering a Grid composed by the nodes of Figure 1, we can imagine to view each node as an agent and we can form the following dependence network $DN = \langle A, G, dep, \geq \rangle$:

1. Agents $A = \{n_1, n_2, n_3, n_4, n_5, n_6\}$;
2. Goals $G = \{g_1, g_2, g_3, g_4, g_5, g_6\}$;
3. $dep(\{n_1\}, \{n_2\}) = \{\{g_1\}\}$: agent n_1 depends on agent n_2 to achieve the goal $\{g_1\}$: to save the file *comp.log*;
   $dep(\{n_2\}, \{n_3\}) = \{\{g_2\}\}$: agent n_2 depends on agent n_3 to achieve the goal $\{g_2\}$: to run the file *mining.mat*;

dep$(\{n_3\}, \{n_1\}) = \{\{g_5\}\}$: agent n_3 depends on agent n_1 to achieve the goal $\{g_5\}$: to save the file *satellite.jpg*;
dep$(\{n_4\}, \{n_6\}) = \{\{g_3\}\}$: agent n_4 depends on agent n_6 to achieve the goal $\{g_3\}$: to run the file *results.mat*;
dep$(\{n_6\}, \{n_5\}) = \{\{g_4\}\}$: agent n_6 depends on agent n_5 to achieve the goal $\{g_4\}$: to save the file *satellite.mpeg*;
dep$(\{n_5\}, \{n_3\}) = \{\{g_6\}\}$: agent n_5 depends on agent n_3 to achieve the goal $\{g_6\}$: to have the authorization to open the file *dataJune.mat*;

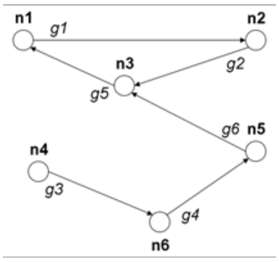


Fig. 4. Dependence Network of Example 6.

Example 6 shows the dependence network based on a simple Grid example composed by six agents. The kind of dependencies are all related to the agent view and they always refer to material goals and not to the institutional ones. This dependence network aims to give a different representation of the system which can be used, for example, for the design of the Grid network. Using dependence networks as methodology to model a system advantage us from different points of view. First, they are abstract, so on the one hand they can be used for example for conceptual modeling, simulation, design and formal analysis. Second, they are used in high level design languages, like TROPOS [10], so they can be used also in software implementation.

4 Dynamic Dependency Modeling

In this section, we answer to the following subquestions: *How to extend dependence networks to build a new modeling activity, called dynamic dependency modeling, able to model the dynamics intrinsic to the notions of the institutional view?* And, *how to model coalitions in dependence networks?*

In multiagent environments, autonomous agents may need to cooperate in order to fulfill their goals. Each group of agents may have different degrees of efficiency in the achievement of its own goals due to differing capabilities of its members. A requirements analysis model has to consider also the possible presence of groups of agents collaborating to each other. We call these groups coalitions. In this section, we introduce the concept of coalition in the NorMAS-RE conceptual metamodel.

4.1 Dynamic Dependence Networks

In Section 3, we introduced the different views composing our conceptual metamodel. On the one hand, we have the agent view where one of the main features is that, since agents are autonomous by definition, no goals and skills can be added to an agent. On the other hand, we have the institutional view where the institutional goals, skills and rules can be added to role, always maintaining the assumption of agents' autonomy. The main changes that can occur thanks to the introduction of the institutional view during the system's evolution are the addition or deletion of an $igoal$, of an $iskill$ and of an $irule$. These additions and deletions change the number of dependencies and the agents involved in them, passing from a dependence network to another one. This change can be represented by means of dynamic dependence networks. We extend Sichman and Conte's [30] theory for conditional dependencies, in which agents can create or destroy dependencies by introducing or removing powers and goals of agents. Goals can be introduced if goals are conditional, or when the agent can create normative goals by creating obligations for the other agents. Otherwise, if an $iskill$ or an $irule$ is introduced, we have the representation of permissions since these additions allow the role to perform a wider number of actions to achieve its goals.

Dependence networks are used to specify early requirements in the TROPOS methodology [10], and to model and reason about the interactions among agents in multiagent systems. Dynamic dependence networks have been firstly introduced by Caire et al. [11] and then treated in Boella et al. [6], in which a dependency between agents depends on the actions of other agents and, in particular, agents can delete the goals of the other ones. Here we distinguish "negative" dynamic dependencies where a dependency exists unless it is removed by a set of agents due to removal of a goal or ability of an agent, and "positive" dynamic dependencies where a dependency may be added due to the power of a third set of agents. *Dynamic dependency modeling* represents the second activity modeling for requirements analysis of the system using NorMAS-RE.

Definition 7 (Dynamic Dependence Networks (DDN)).
A dynamic dependence network is a tuple $\langle A, G, \mathit{dyndep}^-, \mathit{dyndep}^+, \geq \rangle$ where:

- *A is a set of agents;*
- *G is a set of goals;*
- *$\mathit{dyndep}^- : A \times 2^A \times 2^A \rightarrow 2^{2^G}$ is a function that relates with each triple of a agent and two sets of agents all the sets of goals in which the first depends on the second, unless the third deletes the dependency.*
- *$\mathit{dyndep}^+ : A \times 2^A \times 2^A \rightarrow 2^{2^G}$ is a function that relates with each triple of a agent and two sets of agents all the sets of goals on which the first depends on the second, if the third creates the dependency.*
- *$\geq$: $A \rightarrow 2^G \times 2^G$ is for each agent a total pre-order on goals which occur in his dependencies: $G_1 \geq (a) G_2$ implies that $\exists B, C \subseteq A$ such that $a \in B$ and $G_1, G_2 \in \mathit{dyndep}^-(a, B, C)$ or $G_1, G_2 \in \mathit{dyndep}^+(a, B, C)$.*

The static dependencies are defined by $\mathit{dep}(a, B) = \mathit{dyndep}^-(a, B, \emptyset)$.

A graphical representation of the model obtained following the *dynamic dependency modeling* activity is built following the legend of Figure 3 which describes the sign of the dynamic dependency (depicted as a black square) and the dynamic dependency among agents (depicted as one arrowed line connecting two agents with the addition of a label which represents the goal on which there is the dependency and another arrowed dotted line with the sign's label connecting an agent to the arrowed plain line that can be deleted or added by this agent).

Example 7. Considering a Grid composed by the nodes of Figure 1 and the dependence network of Example 6, we can form the following dynamic dependence network $DDN = \langle A, G, \mathit{dyndep}^-, \mathit{dyndep}^+, \geq \rangle$:

1. Agents $A = \{n_1, n_2, n_3, n_4, n_5, n_6\}$;
2. Goals $G = \{g_1, g_2, g_3, g_4, g_5, g_6\}$;
3. $\mathit{dep}(\{n_1\}, \{n_2\}) = \{\{g_1\}\}$: agent n_1 depends on agent n_2 to achieve the goal $\{g_1\}$: to save the file *comp.log*;
   $\mathit{dep}(\{n_2\}, \{n_3\}) = \{\{g_2\}\}$: agent n_2 depends on agent n_3 to achieve the goal $\{g_2\}$: to run the file *mining.mat*;
   $\mathit{dep}(\{n_3\}, \{n_1\}) = \{\{g_5\}\}$: agent n_3 depends on agent n_1 to achieve the goal $\{g_5\}$: to save the file *satellite.jpg*;
   $\mathit{dep}(\{n_4\}, \{n_6\}) = \{\{g_3\}\}$: agent n_4 depends on agent n_6 to achieve the goal $\{g_3\}$: to run the file *results.mat*;
   $\mathit{dep}(\{n_6\}, \{n_5\}) = \{\{g_4\}\}$: agent n_6 depends on agent n_5 to achieve the goal $\{g_4\}$: to save the file *satellite.mpeg*;
   $\mathit{dyndep}^-(n_5, \{n_3\}, \{n_6\}) = \{\{g_6\}\}$: agent n_5 does not depend on agent n_3 to achieve the goal $\{g_6\}$ (to have the authorization to open the file *dataJune.mat*), if it is deleted by agent n_6;
   $\mathit{dyndep}^+(n_5, \{n_4\}, \{n_6\}) = \{\{g_6\}\}$: agent n_5 depends on agent n_4 to achieve the goal $\{g_6\}$ (to have the authorization to open the file *dataJune.mat*), if it is created by agent n_6;

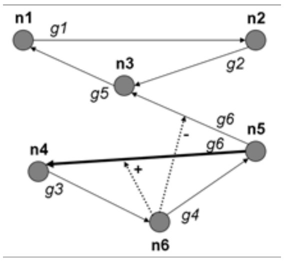


Fig. 5. Dynamic Dependence Network of Example 7.

Example 7 presents the dynamic dependence network of the Grid scenario. We can note that in this network each agent has its associated role since all the nodes are grey ones. Suppose to have a Grid network composing a virtual organization where the local VO administrator is agent n_6. Agent n_6 has delegated the power to give the authorization to access to the files of the VO to agent n_3 but now, since, for example, this node became not safe, the VO administrator has to delegate this power to another node and it chooses node n_4. The dynamic dependence network reflects these actions and thus we have one dynamic dependency for the deletion and another one for the addition.

4.2 Coalitions in Dynamic Dependence Networks

In a multiagent system, we can characterize three different notions of coalitions. A coalition can be defined in dependence networks, based on the idea that to be part of a coalition, every agent has to contribute something, and has to get something out of it. Roughly speaking, a coalition can be formed when there is a cycle of dependencies (the definition of coalitions is more complicated due to the fact that an agent can depend on a set of agents, see below). We show how dependence networks can be used in the requirements analysis for coalitions' evolution, by assuming that goals are maintenance goals rather than achievement goals, which give us automatically a longer term and more dynamic perspective.

A coalition can be represented by a set of dependencies, represented by $C(a, B, G)$ where a is an agent, B is a set of agents and G is a set of goals. Intuitively, the coalition agrees that for each $C(a, B, G)$ part of the coalition, the set of agents B will see to the goal G of agent a. Otherwise, the set of agents B may be removed from the coalition or be sanctioned. The three notions of coalitions defined below make a distinction between the coalitions which cannot be attacked by the others with addition or removal of dynamic dependencies and thus which are actually formed, vulnerable coalitions of which the existence can be destroyed by the deletion of dynamic dependencies and, finally, potential coalitions, those coalitions which can be formed depending on additions and deletions of dynamic dependencies.

Definition 8 (Coalition).
Let A be a set of agents and G be a set of goals. A coalition function is a partial function $C : A \times 2^A \times 2^G$ such that $\{a \mid C(a, B, G)\} = \{b \mid b \in B, C(a, B, G)\}$, the set of agents profiting from the coalition is the set of agents contributing to it.

Let $\langle A, G, dyndep^-, dyndep^+, \geq\rangle$ be a dynamic dependence network, and dep the associated static dependencies.

1. *A coalition function C is a coalition if $\exists a \in A, B \subseteq A, G' \subseteq G$ such that $C(a, B, G')$ implies $G' \in dep(a, B)$. These coalitions which cannot be destroyed by addition or deletion of dependencies by agents in other coalitions.*
2. *A coalition function C is a vulnerable coalition if it is not a coalition and $\exists a \in A, D, B \subseteq A, G' \subseteq G$ such that $C(a, B, G')$ implies $G' \in \cup_D dyndep^-(a, B, D)$. Coalitions which do not need new goals or abilities, but whose existence can be destroyed by removing dependencies.*
3. *A coalition function C is a potential coalition if it is not a coalition or a vulnerable coalition and $\exists a \in A, D, B \subseteq A, G' \subseteq G$ such that $C(a, B, G')$ implies*

$$G' \in \cup_D(dyndep^-(a, B, D) \cup G' \in dyndep^+(a, B, D))$$

   Coalitions which could be created or which could evolve if new abilities or goals would be created by agents of other coalitions on which they dynamically depend.

Example 8. Example 7 presents two different coalitions. On the one hand, we have a *real* coalition composed by agents n_1, n_2 and n_3. On the other hand, we have a potential coalition, such as a coalition which could be formed if agent n_6 really performs the dynamic addition making agent n_5 dependent on agent n_4.

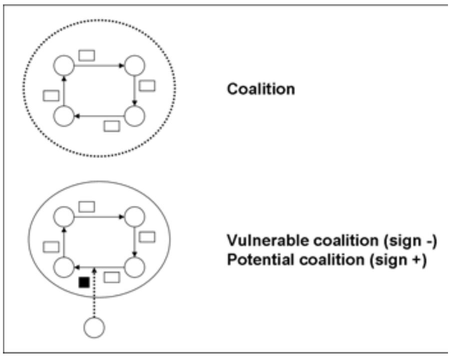


Fig. 6. The legend of the graphical representation of the modeling activities of *dynamic dependency* representing coalitions, potential coalitions and vulnerable coalitions.

These three notions of coalitions represent in the NorMAS-RE model the constraints for coalitions based on the *dynamic dependency modeling*. The graphical representation of the coalition model is depicted in Figure 6 which describes coalitions (depicted as sets of agents and dependencies included in a dotted circle) and vulnerable and potential coalitions (depicted as sets of agents and dependencies in a circle in which one or more of these dependencies can be added or deleted by another agent with a labeled dynamic dependency). There are various further refinements of the notion of coalition. For example, Boella et al. [4] look for minimal coalitions. In this paper we do not consider these further refinements.

5 Conditional Dependency Modeling

In this section, we answer to the subquestions *how to introduce obligations in dependence networks defining a new modeling activity, the conditional dependency modeling* and *how to define new constraints for the coalitions' representation for this new kind of networks*.

Normative multiagent systems are “sets of agents (human or artificial) whose interactions can fruitfully be regarded as norm-governed; the norms prescribe how the agents ideally should and should not behave. [...] Importantly, the norms allow for the possibility that actual behavior may at times deviate from the ideal, i.e., that violations of obligations, or of agents' rights, may occur" [13]. An obligation is a requirement which must be fulfilled to take some course of action, whether legal or moral. The notion of conditional obligation with an associated sanction is the base of the so called regulative norms. Obligations are defined in terms of goals of the agent and both the recognition of the violation and the application of the sanctions are the result of autonomous decisions of the agent. The association of obligations with violations or sanctions is inspired by Anderson's reduction of deontic logic to alethic logic [1].

A well-known problem in the study of deontic logic is the representation of contrary-to-duty structures, situations in which there is a primary obligation and what we might call a secondary obligation, coming into effect when the primary one is violated [23]. A natural effect coming from contrary-to-duty obligations is that obligations pertaining to a particular point in time cease to hold after they have been violated since this violation makes every possible evolution in which the obligation is fulfilled inaccessible. A classical example of contrary-to-duty obligations is given by the so called "gentle murder" by Forrester [18] which says "do not kill, but if you kill, kill gently".

The introduction of norms in dependence networks to present a new modeling activity is based on the necessity to design systems based on norms, particularly obligations. An example of these real applications is due to the introduction of obligations in virtual Grid-based organizations [38] where obligations, as shown in Section 2, are used to enforce the authorization decisions. NorMAS-RE introduces obligations and associates to the violation of these obligations, sanctions and secondary obligations. This is a new design model since, in approaches like [38], obligations are considered simply as tasks that have to be fulfilled when an authorization is accepted/denied while in approaches like [22], the failure in fulfilling the obligation incurs a sanction but there is no secondary obligation.

The first step toward the introduction of obligations directly in dependence networks is to refine the two notions of goal introduced in Section 3. Physical goals are those goals proper of the agent, e.g., in the Grid scenario these are the personal goals of the users of the system, while institutional goals represent those goals associated to a particular role and not to a single agent, e.g., in the Grid scenario, a VO member node has the goal to obtain an authorization to access to a particular file of another node. The introduction of obligations underlines the necessity to introduce a new kind of goal, the normative goals. These goals originate from norms and they represent the obligation itself. We define a new set of normative concepts, based on Boella et al. [3] model of obligations, and we group them in a new view, called the normative view. The normative view is composed by a set of norms N and three main functions, $oblig$, $sanct$ and ctd representing obligations, sanctions and contrary to duty obligations. A portion of the NorMAS-RE metamodel concerning some of the main concepts is shown the UML class diagram of Figure 12.

Definition 9 (Normative View).
Let the agent view $\langle A, F, G, X, goals\colon A \rightarrow 2^G, skills\colon A \rightarrow 2^X, rules: 2^X \rightarrow 2^G\rangle$ and the institutional view $\langle RL, IF, RG, X, igoals: RL \rightarrow 2^{RG}, iskills: RL \rightarrow 2^X, irules\colon 2^X \rightarrow 2^{IF}\rangle$, the normative view is a tuple $\langle A, G, RG, N, oblig, sanct, ctd\rangle$ where:

- *A is a set of agents, G is a set of goals, RG is a set of institutional goals;*
- *N is a set of norms;*

- *the function* $oblig : N \times A \rightarrow 2^{G \cup RG}$ *is a function that associates with each norm and agent, the goals and institutional goals the agent must achieve to fulfill the norm. Assumption:* $\forall n \in N$ *and* $a \in A$, $oblig(n, a) \in power(\{a\})$.
- *the function* $sanct : N \times A \rightarrow 2^{G \cup RG}$ *is a function that associates with each norm and agent the goals and institutional goals that will not be achieved if the norm is violated by agent* a. *Assumption: for each* $B \subseteq A$ *and* $H \in power(B)$ *that* $(\cup_{a \in A} V(n, a)) \cap H = \emptyset$.
- *the function* $ctd : N \times A \rightarrow 2^{G \cup RG}$ *is a function that associates with each norm and agent the goals and institutional goals that will become the new goals the agent has to achieve if the norm is violated by agent* a. *Assumption:*$\forall n \in N$ *and* $a \in A$, $ctd(n, a) \in power(\{a\})$.

Normative goals represent a subset of the union of the personal and institutional goals presented in the agent view and in the institutional view. In Figure 7 the new conceptual metamodel of our model is provided. In this enlarged version of the conceptual metamodel the notions of obligation, sanction and secondary obligation are added.

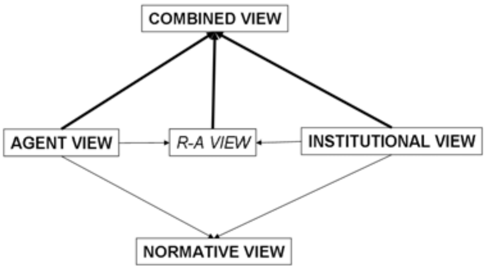


Fig. 7. The new NorMAS-RE conceptual metamodel.

To model obligations, we introduce a set of norms, we associate with each norm the set of agents that has to fulfill it, and for each norm we represent how to fulfill it, and what happens when it is not fulfilled. In particular, we relate norms to goals in the following two ways. First, we associate with each norm n a set of goals and institutional goals $oblig(n) \subseteq G \cup RG$. Achieving these normative goals $oblig(n)$ means that the norm n has been fulfilled; not achieving these goals means that the norm is violated. We assume that every normative goal can be achieved by the group, i.e., the group has the power to achieve it. Second, we associate with each norm a set of goals and institutional goals $sanct(n) \subseteq G \cup RG$ which will not be achieved if the norm is violated (i.e., when the goals resulted from the norm are not achieved) and it represents the sanction associated with the norm. We assume that the group of agents does not have the power to achieve these goals. Third, we associate with each norm (called primary obligation) another norm (called secondary obligation) represented as a set of goals and institutional goals $ctd(n) \subseteq G \cup RG$ that has to be fulfilled if the primary obligation is violated.

Current work on normative systems' formalizations is declarative in nature, focused on the expressiveness of the norms, the definition of formal semantics and the verification of consistency of a given set. Our approach to norms, using the methodology of dependence networks, is different and is based on the definition of conditional dependence networks. Our aim is not to present a new theorem that, using norms semantics, checks whether a given interaction protocol complies with norms. We are more interested in considering, in the context of requirements analysis, how agents' behaviour is effected by norms and in analyzing how to constraint the design of coalitions' evolution thanks to a normative system. There are two main assumptions in our approach. First of all we assume that norms can sometimes be violated by agents in order to keep their autonomy. The violation of norms is handled by sanctions and contrary to duty mechanisms. Second, we assume that, from the institutional perspective, the internal state of the external agents is neither observable nor controllable but the institutional state or public state of the external agents is note since linked to the role associated to the external agent and it can be changed by the agents having this power. Thus, we cannot avoid a forbidden action associated to a goal by a particular rule and we cannot impose an obligatory action in the goals of the agents.

In Section 4, we introduced dynamic dependence networks as a development of the model of dependence networks. In dynamic dependence networks, an agent creates the dependency either creating the obligation, i.e., he creates a new institutional goal for another agent, or creating the power to achieve a goal. In this section, we define a new modeling activity, called *conditional dependency modeling*, to support the early and late requirements analysis of a system representing obligations and, in particular, sanctions and contrary-to-duty obligations. Conditional dependence networks are defined as follows:

Definition 10 (Conditional Dependence Networks (CDN)).
A conditional dependence network is a tuple $\langle A, G, \mathit{cdep}, \mathit{odep}, \mathit{sandep}, \mathit{ctddep}\rangle$ *where:*

- A *is a set of agents;*
- G *is a set of goals;*
- *cdep* $: 2^A \times 2^A \rightarrow 2^{2^G}$ *is a function that relates with each pair of sets of agents all the sets of goals on which the first depends on the second.*
- *odep* $: 2^A \times 2^A \rightarrow 2^{2^G}$ *is a function representing a dependency based on obligations that relates with each pair of sets of agents all the sets of goals on which the first depends on the second.*
- *sandep* $\subseteq (OBL \subseteq (2^A \times 2^A \times 2^{2^G})) \times (SANCT \subseteq (2^A \times 2^A \times 2^{2^G}))$ *is a function relating obligations to the dependency which represent their sanctions. Assumption:* $SANCT \in$ *cdep and* $OBL \in$ *odep.*
- *ctddep* $\subseteq (OBL_1 \subseteq (2^A \times 2^A \times 2^{2^G})) \times (OBL_2 \subseteq (2^A \times 2^A \times 2^{2^G}))$ *is a function relating obligations to the dependency which represent their secondary obligations. Assumption:* $OBL_1, OBL_2 \in$ *odep and* $OBL_1 \cap OBL_2 = \emptyset$*.*

The graphical representation of the model obtained following the *conditional dependency modeling* activity is built following the legend of Figure 8 which describes the obligation-based dependency (depicted as a striped arrowed line), the obligation-based dependency with the associated sanction expressed as conditional dependency (depicted as a striped arrowed line representing the obligation connected to a common arrowed line representing the sanction by a striped line) and the obligation-based dependency with the associated secondary obligation (depicted as a striped arrowed line representing the primary obligation connected to another striped arrowed line representing the secondary obligation by a striped line). The two functions *ctddep* and *sandep* are graphically represented as the striped line connecting the obligation to the sanction or to the secondary obligation.

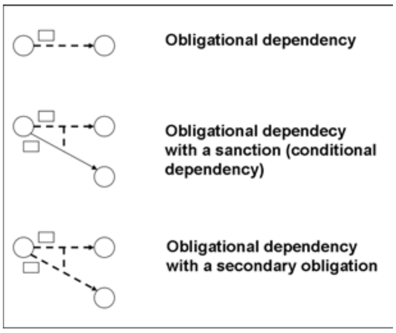

Obligational dependency
Obligational dependecy with a sanction (conditional dependency)
Obligational dependency with a secondary obligation

Fig. 8. The legend of the graphical representation of the modeling activity of *conditional dependency*.

Example 9. Considering Grid's nodes of Example 7, depicted in Figure 5, we can add two constraints for the requirements analysis phase under the form of obligations and we can build the following conditional dependence network $CDN = \langle A, G, \mathit{cdep}, \mathit{odep}, \mathit{sandep}, \mathit{ctddep} \rangle$:

1. Agents $A = \{n_1, n_2, n_3, n_4, n_5, n_6\}$;
2. Goals $G = \{g_1, g_2, g_3, g_4, g_5, g_6, g_7, g_8\}$;
3. $\mathit{cdep}(\{n_1\}, \{n_2\}) = \{\{g_1\}\}$: agent n_1 depends on agent n_2 to achieve the goal $\{g_1\}$: to save the file *comp.log*;
   $\mathit{dep}(\{n_2\}, \{n_3\}) = \{\{g_2\}\}$: agent n_2 depends on agent n_3 to achieve the goal $\{g_2\}$: to run the file *mining.mat*;
   $\mathit{dep}(\{n_3\}, \{n_1\}) = \{\{g_5\}\}$: agent n_3 depends on agent n_1 to achieve the goal $\{g_5\}$: to save the file *satellite.jpg*;
   $\mathit{dep}(\{n_4\}, \{n_6\}) = \{\{g_3\}\}$: agent n_4 depends on agent n_6 to achieve the goal $\{g_3\}$: to run the file *results.mat*;
   $\mathit{dep}(\{n_6\}, \{n_5\}) = \{\{g_4\}\}$: agent n_6 depends on agent n_5 to achieve the goal $\{g_4\}$: to save the file *satellite.mpeg*;
   $\mathit{dep}(\{n_5\}, \{n_4\}) = \{\{g_6\}\}$: agent n_5 depends on agent n_4 to achieve the goal $\{g_6\}$: to have the authorization to open the file *dataJune.mat*;
   $\mathit{odep}(\{n_2\}, \{n_1\}) = \{\{g_7\}\}$: agent n_2 is obliged to perform goal $\{g_7\}$ concerning agent n_1 : to run the file *mining.mat* with the highest priority;
   $\mathit{odep}(\{n_4\}, \{n_5\}) = \{\{g_8\}\}$: agent n_4 is obliged to perform goal $\{g_8\}$ concerning agent n_5 : to share results of the running of file *dataJune.mat* with agent n_5;

odep$(\{n_4\},\{n_6\}) = \{\{g_8\}\}$: agent n_4 is obliged to perform goal $\{g_8\}$ concerning agent n_6 : to share results of the running of file *dataJune.mat* with agent n_6;
sandep$\{((\{n_2\},\{n_1\}) = \{\{g_7\}\}, (\{n_1\},\{n_2\}) = \{\{g_1\}\})\}$;
ctddep$\{((\{n_4\},\{n_5\}) = \{\{g_8\}\}, (\{n_4\},\{n_6\}) = \{\{g_8\}\})\}$;

Example 9 shows the subsequent step after the deletion and the insertion of the two dynamic dependencies of Example 7. In this situation, following the definition of coalition, we can imagine to have two local coalitions composing a virtual organization, the first one composed by nodes n_1, n_2, n_3 and the other composed by nodes n_4, n_5 and n_6. Since these two subsets of the virtual organization have to work with a good cohesion then it is possible to insert some constraints, made clear by obligations. The first obligation consists in giving the highest priority to, for example, a computation for an agent composing the same local coalition as you. This first obligation is related to a sanction if it is violated. This link is made clear by the function *sandep* and it represents the deletion of a dependence concerning a goal of the agent that has to fulfill the obligation. The second obligation, instead, is related to a secondary obligation and it means that the agent has to share the results of a computation with a member of its local coalition but, if it does not fulfill this obligation then it has to share these results with another member of the local coalition.

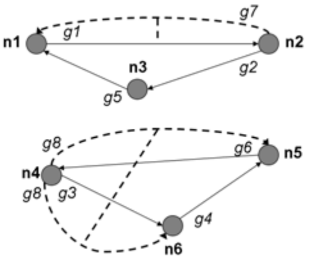


Fig. 9. Conditional Dependence Network of Example 9.

In this new kind of network, if a goal, set by an obligation, is not fulfilled then the conditional dependency related to this obligation has two possible developments: if a sanction is associated to the norm, a goal cannot be achieved and thus the conditional dependency related to that goal has to be deleted or, if a contrary-to-duty obligation, which means a secondary obligation, is associated to the norm then the conditional dependency related the goals set by this secondary obligation has to be added. We represent obligations, sanctions and contrary-to-duty obligations as tuples of dependencies related to each other. An obligation is viewed as a particular kind of dependency and it is related to other dependencies: dependencies due to sanctions and dependencies due to secondary obligations. In the first case, we have that sanctions are common dependencies, already existing inside the system that, because of their connection with the obligation, can be deleted. In particular, if the obligation is not fulfilled, then the dependency related to the obligation with the role to be its sanction is deleted. In the second case, instead, a primary obligation is related to a number of secondary obligations. A graphical representation of the evolutions of conditional dependence networks is provided in Figure 10:

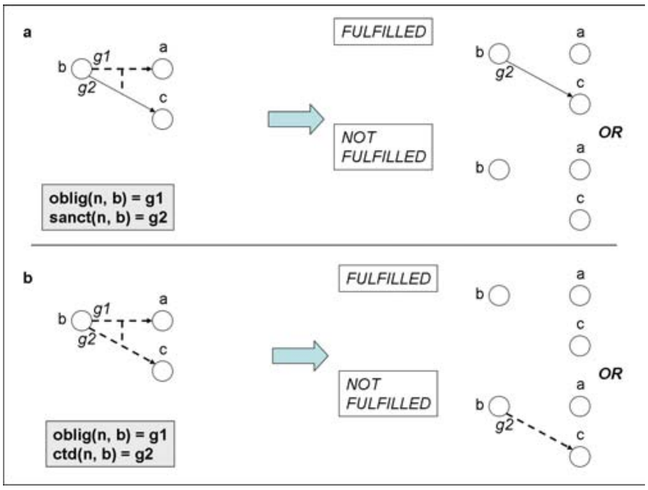


Fig. 10. The evolution of conditional dependence networks.

In the first case, if the obligation is fulfilled and it is linked to a sanction then the obligation can be removed and also the connection among the obligation and the sanction. The only dependency that remains in the network is the one related to the sanction that passes from being a conditional dependency to a common dependency. If the obligation is not fulfilled then it is deleted and the deletion involves also the conditional dependency representing the sanction. The sanction consists exactly the deletion of this conditional dependency. In the second case, if the obligation is fulfilled and it is linked to a secondary obligation then the obligation is deleted and also the secondary obligation is deleted since there is no reason to already exists. If the obligation, instead, is not fulfilled then the primary obligation is deleted but the secondary obligation not. Note that in Figure 10 are depicted only the conditional dependencies and the obligational dependencies and not all the other kinds of possible dependencies of the network.

Two case studies: transactions and personal norms. In this section, we analyze two particular case studies using our representation of obligations. The first one consists in transactions. A transaction is an agreement or communication carried out between separate entities, often involving the exchange of items of value, such as information, goods, services and money. This is the basic idea underlying norm emergence. Let us consider the case of two agents a and b, where a is the buyer and b is the seller. If we consider two goals such as $g1$: book sent by the seller b to the buyer a and $g2$: money transferred from the buyer a to the seller b, we have the dependence network depicted in Figure 11-(b). The two agents depend on each other to achieve their goals, the seller is waiting for its payment and the buyer is waiting for its good. When introduced, our representation of obligations allows to arrive to a very simplified version of the network in which each agent depends on itself to not violate the obligation. The dependence network derived after the norm creation is much more simpler than the previous one representing however the same concepts. This simplified version of the network, representing obligations, can be used for the design and, in particular for the requirements analysis phases, of the multiagent system allowing to individuate in a simpler way the obligation present in it, without the necessity to take into account all the sets of dependencies on goals of the network.

The second case study makes more explicit this necessity to simplify the dependence network with the aim to individuate the obligations is the case of personal norms. In the real life, everybody's life is regulated by personal norms like *not kill* and *not leave trash on the roads*. These norms are referred to every person and it seems that everyone depends on the others to achieve these goals that can be represented as goals of the whole society. It is similar to the social delegation cycle: do not do the others what you do not want them to do to you. In this case, we can represent the dependence network as a full connected graph since every agent depends on all the other agents, for example to not be killed. The simplification brought by the representation of obligations is relevant, as can be seen in Figure 11-(a).

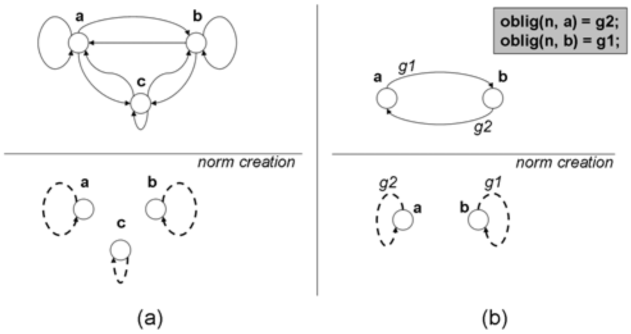


Fig. 11. Case studies: personal norms and transactions.

5.1 Coalitions in Conditional Dependence Networks

In this Section, we answer to the subquestion: *What constrains are set by obligations to the conditional dependency modeling concerning coalitions*. In Section 4, we presented three different kinds of coalitions: existing coalitions composed by common dependencies, vulnerable coalitions composed by one or more arcs linked to a dynamic dependency of removal and, finally, potential coalitions composed by one or more arcs linked to a dynamic dependency of addition. The new kind of dependence networks, conditional dependence networks, has to be taken into account when a system is described in terms of coalitions. This means that coalitions, vulnerable coalitions and potential coalitions can change depending on the conditional dependencies set by the obligations of the system. A coalition has to consider also sanctions and secondary obligations, according to these constraints:

Definition 11 (Constraints for Conditional Dependency Modeling). *Let A be a set of agents and G be a set of goals. A coalition function is a partial function $C \subseteq A \times 2^A \times 2^G$ such that $\{a \mid C(a, B, G)\} = \{b \mid b \in B, C(a, B, G)\}$, the set of agents profiting from the coalition is the set of agents contributing to it.*

Introducing conditional dependence networks, the following constraints arise:

- *$\forall(dep_1, dep_2) \in$ sandep, $dep_2 \notin C$ if and only if $dep_1 \notin C$. If the obligation, associated to the dependency dep_1 is not part of the coalition C then also the sanction dep_2 associated to the obligation is not part of the coalition C. If the obligation, associated to the dependency dep_1 is part of the coalition C then also the sanction dep_2 associated to the obligation is part of the coalition C.*
- *$\forall(dep_1, dep_2) \in$ ctddep, $dep_2 \in C$ if and only if $dep_1 \notin C$. If the primary obligation, associated to the dependency dep_1 is not part of the coalition C then the secondary obligation dep_2 is part of the coalition C. If the primary obligation, associated to the dependency dep_1 is part of the coalition C then the secondary obligation dep_2 is not part of the coalition C.*

Example 10. Let us consider the conditional dependence network of Example 9. Applying these constraints, we have that if the obligation on goal g_7 is fulfilled then the local coalition composed by agents n_1, n_2 and n_3 already exists since the dependency associated to the sanction is not deleted. If the obligation on goal g_7 is not fulfilled then the obligation is deleted but also the sanction is deleted and the coalition does not exist any more. Concerning the second local coalition, if the obligation is fulfilled then both the primary and the secondary obligation are removed but if the primary obligation is not fulfilled then the secondary obligation is part of the local coalition composed by agents n_4, n_5 and n_6.

5.2 Regulation of Stability

In game theoretical approaches [27], stability may be taken into account when distributing the payoff of the coalition among its members. Roughly speaking, payoffs should be divided in a fair way to maintain stability. The core, for example, provides a concept of stability for coalitional games and a payoff is in the core only if no coalition has an incentive to break off from the grand coalition and form its own group. Other approaches of the same kind are provided by the other solution concepts such as the Shapley value and the nucleous. Given a previously formed coalitional configuration, game theory usually concentrates on checking its stability or its fairness and on the calculation of the corresponding payments. But game theory rarely takes into consideration the special properties of a multi-agent environment such as, for example, goal-based agents. Coalitions change dynamically due to rapid changes in the tasks and resource availability, and therefore relying on the initial configurations is misleading.

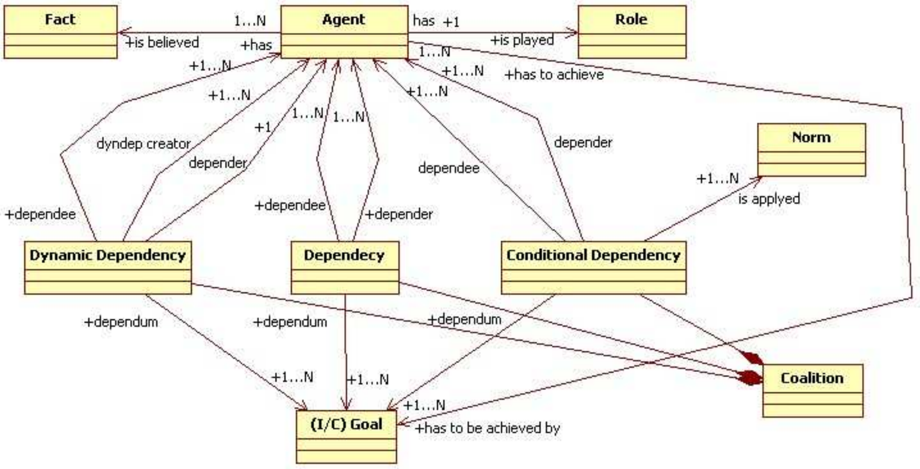


Fig. 12. The UML class diagram specifying the main concepts of the NorMAS-RE metamodel.

In this section, we present a first step toward the definition of a notion of stability for coalitions individuated in the context of one of our three modeling activities. The importance of the definition of a notion of stability for the modeling analysis, particularly for the requirements analysis phases, is related to the issues of security and efficiency. For example, in the Grid scenario, it is very important to have the guarantee that the two subsets composing the virtual organization are stable in the sense that they represent secure and efficient “group" of nodes with a great internal cohesion. This approach has the aim to present the problem of coalitions’ stability from a different point of view respect the point of view presented in game theoretical approaches. The main difference is in the notion of agent used in the NorMAS-RE model, such as not an agent viewed only as a utility maximizer but a goal-based agent. In this sense, our definition of agent is more complex and with many facets than the agents presented in game theory. Starting from Section 4, where we distinguished among three different kinds of coalitions, we can start to define coalitions’ stability in the following way:

Definition 12 (Coalitions’ stability). *A coalition C is called stable if $\exists a \in A, B \subseteq A, G' \subseteq G$ such that $C(a, B, G')$ implies $G' \in$ dep(a, B) and $\neg(\exists a \in A, D, B \subseteq A, G' \subseteq G$ such that $C(a, B, G')$ implies $G' \in \cup_D$dyndep$^-(a, B, D))$. A coalition is stable if it is formed by dependencies relying its members and there is not the possibility to delete one these dependencies by another agent, inside or outside the coalition itself.*

Conditional dependencies add new possibility to see to the stability of a coalition. In fact, we can claim that one of the main interests of the agents involved in a coalition is to maintain its stability. This maintenance can be achieved using norms such as obligations to regulate the behaviour of the members of the coalition. The use of obligations can follow two different lines:

- Obligations to regulate dynamic dependencies: this first kind of obligation is addressed to each member of a coalition with the aim to avoid, imposing a sanction or a secondary obligation, the mining of the stability of a coalition. Following the notion of stability, the first norm of all the agents when they become members of a coalition is informally: *If an agent, member of a coalition, has the power to delete one or more of the dependencies constituting the coalition itself then it is obliged to not do this deletion.* This norm is addressed only to those agent belonging to the coalition since, as in real cases, it is always possible for an external agent or coalition to attack another coalition with the aim to decrease its influence. This obligation can be linked to sanctions and secondary obligations of different kinds, such as for example the secondary obligation to create another dependency with the aim to strengthen the coalition. It is also possible to impose a sanction to the agent, for example deleting all the dependencies in which it depends on other agents, preventing him to achieve its goals.
- Obligations to regulate agents' behaviour: this second kind of obligations is not related to the dependencies and dynamic dependencies describing the system, but it is addressed to the regulation of the behaviour of the agents depending on their membership to a coalition. These kind of obligations are of the type *If an agent belongs to a coalition then it has to satisfy first those requests coming from the other members of the coalition and, only after, requests coming from outsiders.* These rules aim to strengthen the unity of the coalition and to improve the work inside it.

6 Related work

The related work section is divided into three main sections with the aim to follow better the different research lines along which the paper is developed. The three sections consists in 1) works on agent-based software engineering, 2) works on coalition formation and coalitions' evolution taking into account both game theoretical approaches and social networks ones, 3) works on normative multiagent systems and institutions. The second section presents also a number of works devoted to the definition of the notion of stability for coalitions.

6.1 Agent-based software engineering

The idea of focusing the activities that precede the specification of software requirements, in order to understand how the intended system will meet organizational goals, is not new. It has been first proposed in requirements engineering, specifically in Eric Yu's work with his i* model [36]. This model has been applied in various application areas, including requirements engineering, business process re-engineering, and software process modeling. The i* model offers actors, goals and actor dependencies as primitive concepts [35]. The rationale of the i* model is that by doing an earlier analysis, one can capture not only the what or the how, but also the why a piece of software is developed. This, in turn, supports a more refined analysis of system dependencies and encourages a uniform treatment of the system's functional and non-functional requirements. As stated in the introduction and in the paper, the most important example for our model consists in the TROPOS methodology [10] that aspires to span the overall software development process, from early requirements to implementation. Other approaches to software engineering are those of KAOS [15] which covers only the late requirements phase, GAIA [34] which covers both the late requirements phase and the architectural design, AAII [21] and MaSE [16] which cover the two phases of architectural and detailed design, and AUML [2] which covers only the detailed design. The main difference between these approaches and our approach is in the use at the same time of the normative multiagent paradigm based on both the notion of institution and the notion of obligation, the graphical modeling language based on dependencies among agents and the covering of the very early phases of requirements analysis.

6.2 Coalitions' formation and evolution

One of the most important issues in the field of multiagent systems concerns the description and formalization of coalition formation. Although there were many approaches defining coalition formation, to represent different perspectives. Two representative examples are given by the model of Shehory and Kraus [26] and the one of Sichman and Conte [29][30]. The approach of Shehory and Kraus [26] is based on the assumption that autonomous agents in the multiagent environments may need to cooperate in order to fulfill tasks. They present algorithms that enable the agents to form groups and assign a task to each group, calling these groups coalitions. However, Shehory and Kraus' work considers tasks which are not related to the individual goals of the agents in the coalition and it does not consider the motivations for agents to enter the coalition, nor the dependencies existing among the agents. They only address cases in which dependencies among tasks are due to competing resources' requirements or execution precedence order. Sichman [29], instead, introduces a different point of view. He presents coalition formation using a dependence-based approach based on the notion of social dependence introduced by Castelfranchi [14]. This model introduces the notion of dependence situation, which allows an agent to evaluate the susceptibility of other agents to adopt his goals, since agents are not necessarily supposed to be benevolent and therefore automatically adopt the goals of each other. In this dependence-based model, coalitions can be modeled using dependence networks. A definition of coalitions inspired by dependence networks is given by Boella et al. [4]. The authors represent a potential coalition as a labeled AND-graph of dependencies among agents. These AND-graphs consist of a set of nodes which denotes the agents involved in the coalition and a set of labeled arcs.

Coalitions' stability The work that, to our knowledge, gives a first definition of stability is the paper of Zlotkin and Rosenschein [39]. In a task oriented domain, a coalition can coordinate by redistributing their tasks among themselves. It seems intuitively reasonable that agents in a coalition game should not suffer by coordinating their actions with a larger group. In other words, if you take two disjoint coalitions, the utility they can derive together should not be less than the sum of their separate utilities, at the worst, they could coordinate by ignoring each other. This property is called superadditivity. This work introduce the notion of stability of a coalition using the concept of superadditivity. The stability condition relates to the payoff vector that assigns to each agent a utility. There are three levels of stability conditions: individual, group and coalition rationality. Individual rationality means that no individual agent would like to opt out of the full coalition, group rationality means that the group as a whole would not prefer any other payoff vector over this vector and coalition rationality means that no group of agents should have an incentive to deviate from the full coalition and create a subcoalition for each subset of agents. To ensure stability, they need to find a consensus mechanism that is resistant to any coalition manipulation. Another work on this issue is from Sandholm and Lesser [24]. In this paper, the optimal coalition structure and its stability are significantly affected by the agents algorithms performance profiles and the unit cost of computation.

6.3 Normative multiagent systems and institutions

An example of normative multiagent system introducing obligations has been done by Boella and van der Torre [7]. In this work, to model obligations they introduce a set of norms, associated with each norm the set of agents that has to fulfill it and what happens when it is not fulfilled. In particular, they relate norms to goals in the following two ways. First, each norm is associated to a set of goals. Achieving these normative goals means that the norm has been fulfilled; not achieving these goals means that the norm is violated. They assume that every normative goal can be achieved by the group, that means that the group has the power to achieve it. The second point is that each norm is associated to another set of goals which will not be achieved if the norm is violated, this is the sanction associated to the norm. They assume that the group of agents does not have the power to achieve these goals, otherwise they would avoid the sanction.

An interesting approach to the application of the notion of institution to multiagent systems is defined in Sierra et al. [31]. Electronic Institutions (EIs) provide the virtual analogue of human organizations in which agents, playing different organizational roles, interact to accomplish individual and organizational goals. EIs introduce sets of artificial constraints that articulate and coordinate interactions among agents. In this approach, roles are defined as patterns of behavior and are divided into institutional roles (those enacted to achieve and guarantee institutional rules) and non-institutional roles (those requested to conform to institutional rules). Like us, the purpose of their normative rules is to affect the behavior of agents by imposing obligations or prohibitions.

Another approach to EIs is given by Bogdanovych et al. [8]. In this approach they propose the use of 3D Virtual Worlds to include humans into software systems with a normative regulation of interactions. The normative part can be seen as defining which actions require an institutional verification assuming that any other action is allowed. Inside the 3D Interaction Space, an institution is represented as a building where the participants are represented as avatars. Once they enter the building their actions are validated against the specified institutional rules. In the last two works, unlike us, the concept of institution is presented by a practical approach without a formal definition of the concept of institution and a description of its dynamics while they are similar to our one in the establishment of a different level of the organization related to the institution.

The problem of dynamic institutions is treated in Bou et al. [9] as an extension to EIs definition with the capability to decide in an autonomous way how to answer dynamically to changing circumstances through norm adaptation and changes in institutional agents. The assumption for EIs to adapt is that EIs seek specific goals. The paper presents the normative transition function that maps a set of norms into another one. As our approach, agents participating in the system have social interactions mediated by the institution and the consequences of these interactions is a change in the institutional state of an agent. The difference with our approach consists in the definition of the institution as an entity with own goals, the running example given into the paper is that of the institution of the Traffic Regulation Authority with the goal to decrease the number of accidents below a given threshold, and states.

An interesting approach is presented in Vazquez-Salceda et al. [33] where they propose the Organizational Model for Normative Institutions (OMNI) framework. OMNI brings together some aspects from two existing frameworks: OperA and HARMONIA. OperA is a formal specification framework that focuses on the organizational dimension while HARMONIA is a formal framework to model especially highly regulated electronic organizations from an abstract level to the final protocols that implement norms. In OMNI, roles are often dependent on other roles for the realization of their objectives. Societies establish dependencies and power relations between roles, indicating relationships between roles. These relationships describe, like in our approach, how actors can interact and contribute to the realization of the objectives of each other.

7 Conclusions

This paper provides a detailed account of NorMAS-RE, a new requirements analysis model based on the normative multiagent paradigm, following the TROPOS methodology [10]. The paper presents and discusses the early and late requirements phases of systems design. The first part of the paper presents the key concepts of our model dividing them into three submodels, one representing the agents and their mentalistic notions of goals and facts, the second representing the roles and their associated notions of institutional goals and facts and, finally, the third representing the mapping between agents and roles. The second part of the paper presents our graphical representations for the three modeling activities by which NorMAS-RE is composed. The three modeling activities are called *dependency modeling*, *dynamic dependency modeling* and *conditional dependency modeling* and they are based on the notions of institution, obligation, sanction and secondary obligation. The addition of normative concepts as the last ones is a relevant improvement to requirements analysis since it allows first to constraint the construction of the requirements modeling and second to represent systems, as in the Grid scenario, in which there are explicit obligations regulating the behaviour of the components composing it. Moreover, the NorMAS-RE model is defined also to model the requirements analysis phases in a context in which there is the possible presence of coalitions and we present the first step toward the definition of the notion of coalitions' stability for our modeling activities.

Our long term objective is to provide a detailed account of the NorMAS-RE model and to start the development of the other phases of design analysis of a system such as architectural and detailed design phases and the implementation phase, as done for example by the TROPOS methodology. Moreover, the NorMAS-RE model in its current form is also not suitable for agents requiring advanced reasoning mechanisms for plans, goals and negotiations. Further extensions will be required to the NorMAS-RE model to address this class of software applications.

References

1. A. Anderson. The logic of norms. *Logic et analyse*, 2, 1958.
2. B. Bauer, J. P. Müller, and J. Odell. Agent uml: A formalism for specifying multiagent software systems. *International Journal of Software Engineering and Knowledge Engineering*, 11(3):207–230, 2001.
3. G. Boella, P. Caire, and L. van der Torre. Autonomy implies creating one's own norms norm negotiation in online multi-player games. *KAIS*, 2008.
4. G. Boella, L. Sauro, and L. van der Torre. Strengthening admissible coalitions. In *ECAI 2006*, pages 195–199, 2006.
5. G. Boella, L. van der Torre, and H. Verhagen. Introduction to normative multiagent systems. *Computational and Mathematical Organization Theory*, 12:71–79, 2006.
6. G. Boella, L. van der Torre, and S. Villata. Social viewpoints for arguing about coalitions. In The Duy Bui, Tuong Vinh Ho, and Quang-Thuy Ha, editors, *PRIMA*, volume 5357 of *Lecture Notes in Computer Science*, pages 66–77. Springer, 2008.
7. Guido Boella and Leendert W. N. van der Torre. Power in norm negotiation. In *KES-AMSTA*, pages 436–446, 2007.
8. A. Bogdanovych, M. Esteva, S. Simoff, C. Sierra, and H. Berger. A methodology for developing multiagent systems as 3d electronic institutions. In *Proceedings of AOSE@AAMAS'07*, 2007.
9. E. Bou, Lopez-Sanchez M., and Rodriguez-Aguilar J. A. Adaptation of automatic electronic institutions through norms and institutional agents. *Engineering Societies in the Agents World VII*, 2007.
10. P. Bresciani, A. Perini, P. Giorgini, F. Giunchiglia, and J. Mylopoulos. Tropos: An agent-oriented software development methodology. *Autonomous Agents and Multi-Agent Systems Journal*, 8:203–236, 2004.
11. P. Caire, S. Villata, L. van der Torre, and G. Boella. Conviviality masks in role-based institutions multi-agent teleconferencing in virtual worlds. In *Proceedings of AAMAS'08*, 2008.
12. K. M. Carley. Dynamic network analysis. In *Dynamic Social Network Modeling and Analysis: Workshop Summary and Papers*, pages 133–145, 2003.
13. J. Carmo and A.J.I. Jones. Deontic logic and contrary-to-duties. *Handbook of Philosophical Logic*, pages 203–279, 1996.
14. C. Castelfranchi. The micro-macro constitution of power. *Protosociology*, 18:208–269, 2003.
15. A. Dardenne, A. van Lamsweerde, and S. Fickas. Goal-directed requirements acquisition. *Sci. Comput. Program.*, 20(1-2):3–50, 1993.
16. S. DeLoach. Analysis and design using mase and agent tool. In *MAICS 2001*, 2001.
17. J. Ferber, O. Gutknecht, and F. Michel. Fom agents to organizations: An organizational view of multi-agent systems. In *Proceedings of AOSE '03*, pages 214–230, 2003.
18. J. W. Forrester. Gentle murder, or the adverbial samaritan. *Journal of Philosophy*, 81:193–197, 1984.
19. D. Grossi. *Desigining Invisible Hancuffs. Formal Investigations in Institutions and Organizations for Multi-agent Systems.* SIKS Dissertation Series 2007-16, PhD Thesis, 2007.

20. A. J. I. Jones and M. Sergot. A formal characterization of institutionalised power. *Logic Journal of IGPL*, 2003.
21. D. Kinny, M. P. Georgeff, and A. S. Rao. A methodology and modelling technique for systems of bdi agents. In W. Van de Velde and J. W. Perram, editors, *MAAMAW*, volume 1038 of *Lecture Notes in Computer Science*, pages 56–71. Springer, 1996.
22. N. H. Minsky and A. Lockman. Ensuring integrity by adding obligations to privileges. In *ICSE*, pages 92–102, 1985.
23. H. Prakken and M. Sergot. Contrary-to-duty obligations. *Studia Logica*, 1996.
24. T. W. Sandholm and V. R. Lesser. Coalition formation among bounded rational agents. Technical report, CMPSCI, 1995.
25. L. Sauro. *Formalizing admissibility criteria in coalition formation among goal directed agents*. PhD thesis, University of Turin, 2005.
26. O. Shehory and S. Kraus. Methods for task allocation via agent coalition formation. *Artificial Intelligence*, 101:165–200, 1998.
27. Y. Shoham and K. Leyton-Brown. *Multiagent Systems: Algorithmic, Game-Theoretic, and Logical Foundations*. Cambridge University Press, 2008.
28. Y. Shoham and M. Tennenholtz. On social laws for artificial agent societies: off-line design. *Artificial Intelligence*, pages 231–252, 1995.
29. J. S. Sichman. Depint: Dependence-based coalition formation in an open multi-agent scenario. *Artificial Societies and Social Simulation*, 1(2), 1998.
30. J. S. Sichman and R. Conte. Multi-agent dependence by dependence graphs. In *AAMAS'02*, pages 483–490, 2002.
31. C. Sierra, J. A. Rodriguez-Aguilar, P. Noriega, J. L. Arcos, and M. Esteva. Engineering multi-agent systems as electronic institutions. *European Journal for the Informatics Professional*, 2004.
32. R. O. Sinnott, D. W. Chadwick, T. Doherty, D. Martin, A. Stell, G. Stewart, L. Su, and J. P. Watt. Advanced security for virtual organizations: The pros and cons of centralized vs decentralized security models. In *CCGRID*, pages 106–113. IEEE Computer Society, 2008.
33. J. Vázquez-Salceda, V. Dignum, and F. Dignum. Organizing multiagent systems. *Journal of Autonomous Agents and Multi-Agent Systems*, 11(3):307–360, 2005.
34. M. Wooldridge, N. R. Jennings, and D. Kinny. The gaia methodology for agent-oriented analysis and design. *Autonomous Agents and Multi-Agent Systems*, 3(3):285–312, 2000.
35. E. Yu. Modeling organizations for information systems requirements engineering. In *First IEEE International Symposium on Requirements Engineering*, pages 34–41, 1993.
36. E. Yu. *Modelling Strategic Relationships for Process Reengineering*. PhD thesis, University of Toronto, 1995.
37. F. Zambonelli, N. Jennings, and M. Wooldridge. Developing multiagent systems: The gaia methodology. *IEEE Transactions of Software Engineering and Methodology*, 12:317Ű370, 2003.
38. G. Zhao, D. W. Chadwick, and S. Otenko. Obligations for role based access control. In *AINA Workshops (1)*, pages 424–431. IEEE Computer Society, 2007.
39. G. Zlotkin and J. S. Rosenschein. Coalition, cryptography, and stability, mechanisms for coalition formation in task oriented domains. In *AAAI 1994*, pages 432–437, 1994.

Normative Multi-Agent Programs and Their Logics

Mehdi Dastani[1], Davide Grossi[2], John-Jules Ch. Meyer[1], and Nick Tinnemeier[1]

[1] Universiteit Utrecht
The Netherlands
[2] Computer Science and Communication
University of Luxembourg, Luxembourg

Abstract. Multi-agent systems are viewed as consisting of individual agents whose behaviors are regulated by an organization artefact. This paper presents a simplified version of a programming language that is designed to implement norm-based artefacts. Such artefacts are specified in terms of norms being enforced by monitoring, regimenting and sanctioning mechanisms. The syntax and operational semantics of the programming language are introduced and discussed. A logic is presented that can be used to specify and verify properties of programs developed in this language.

1 Introduction

In this paper, multi-agent systems are considered as consisting of individual agents that are autonomous and heterogenous. Autonomy implies that each individual agent pursues its own objectives and heterogeneity implies that the internal states and operations of individual agents may not be known to external entities [14, 7]. In order to achieve the overall objectives of such multi-agent systems, the observable/external behavior of individual agents and their interactions should be regulated/coordinated.

There are two main approaches to regulate the external behavior of individual agents. The first approach is based on coordination artefacts that are specified in terms of low-level coordination concepts such as synchronization of processes[12]. The second approach is motivated by organizational models, normative systems, and electronic institutions[13, 10, 7, 8]. In such an approach, norm-based artefacts are used to regulate the behavior of individual agents in terms of norms being enforced by monitoring, regimenting and sanctioning mechanisms. Generally speaking, the social and normative perspective is conceived as a way to make the development and maintenance of multi-agent systems easier to manage. A plethora of social concepts (e.g., roles, social structures, organizations, institutions, norms) has been introduced in multi-agent system methodologies (e.g. Gaia [14]), models (e.g. OperA [6], $\mathcal{M}$oise$^+$ [9], electronic institutions and frameworks (e.g. AMELI [7], $\mathcal{S}$-$\mathcal{M}$oise$^+$ [9]).

The main contribution of this paper is twofold. On the one hand, a simplified version of a programming language is presented that is designed to implement

Dagstuhl Seminar Proceedings 09121
Normative Multi-Agent Systems
http://drops.dagstuhl.de/opus/volltexte/2009/1907

multi-agent systems in which the observable (external) behavior of individual agents is regulated by means of norm-based artefacts. Such artefacts are implemented in terms of social concepts such as norms and sanctions, monitor the actions performed by individual agents, evaluate their effects, and impose sanctions if necessary. On the other hand, we devise a logic to specify and verify properties of programs that implement norm-based artefacts.

In order to illustrate the idea of norm-based artefacts, consider the following simple example of a simulated train station where agents ought to buy a ticket before entering the platform or trains. To avoid the queue formation, agents are not checked individually before allowing them to enter the platform or trains. In this simulation, being on the platform without a ticket is considered as a violation and getting on the train without having a ticket is considered as a more severe violation. A norm-based artefact detects (all or some) violations by (all or some) agents and reacts on them by issuing a fine if the first violation occurs, for instance by charging the credit card of the defecting user, and a higher fine if the second violation occurs.

In this paper, we first briefly explain our idea of normative multi-agent systems and discuss two norm-based approaches to multi-agent systems, that is, ISLANDER/AMELI [7] and S-MOISE+ [9]. In section 3, we present the syntax and operational semantics of a programming language designed to implement normative multi-agent systems. This programming language allows the implementation of norm-based artefacts by providing programming constructs to represent norms and mechanisms to enforce them. In section 4, a logic is presented that can be used to specify and verify properties of norm-based artefacts implemented in the presented programming language. Finally, in section 5, we conclude the paper and discuss some future directions in this research area.

2 Norms and Multi-Agent Systems

Norms in multi-agent systems can be used to specify the standards of behavior that agents ought to follow to meet the overall objectives of the system. However, to develop a multi-agent system does not boil down to state a number of standards of behavior in the form of a set of norms, but rather to organize the system in such a way that those standards of behavior are actually followed by the agents. This can be achieved by regimentation [10] or enforcement mechanisms, e.g., [8].

When regimenting norms all agents' external actions leading to a violation of those norms are made impossible. Via regimentation (e.g., gates in train stations) the system prevents an agent from performing a forbidden action (e.g., entering a train platform without a ticket). However, regimentation drastically decreases agent autonomy. Instead, enforcement is based on the idea of responding after a violation of the norms has occurred. Such a response, which includes sanctions, aims to return the system to an acceptable/optimal state. Crucial for enforcement is that the actions that violate norms are observable by the system (e.g., fines can be issued only if the system can detect travelers entering the platform or trains without a ticket). Another advantage of having enforcement over regimentation is that allowing for violations contributes to the flexibility and autonomy of the agent's behavior [3]. These norms are often specified by means of concepts like permissions, obligations, and prohibitions.

In the literature of multi-agent systems related work can be found on electronic institutions. In particular, ISLANDER[7] is a formal framework for specifying norms in institutions, which is used in the AMELI platform [7] for executing electronic institutions based on norms provided in it. However, the key aspect of ISLANDER/AMELI is that norms can never be violated by agents. In other words, systems programmed via ISLANDER/AMELI make only use of regimentation in order to guarantee the norms to be actually followed. This is an aspect which our approach intends to relax guaranteeing higher autonomy to the agents, and higher flexibility to the system.

A work that is concerned with programming multiagent systems using (among others) normative concepts is also S-MOISE+, which is an organizational middleware that follows the Moise+ model[9]. This approach, like ours, builds on programming constructs investigated in social and organizational sciences. However, S-MOISE+ lacks formal operational semantics, which is instead the main contribution of the present paper to the development of programming languages form multi-agent systems. Besides, norms in S-MOISE+ typically lack monitoring and sanctioning mechanisms for their implementation which are, instead, the focus of our proposal. It should be noted that [11] advocates the use of artifacts to implement norm enforcement mechanisms. However, it is not explained how this can be done using those artifacts.

To summarize, ISLANDER/AMELI implements norm via full regimentation, while in S-MOISE+ violations are possible, although no specific system's response to violations is built in the framework. We deem these shortcomings to have a common root, namely the absence of a computational model of norms endowed with a suitable operational semantics. The present paper fills this gap along the same lines that have been followed for the operationalization of BDI notions in the APL-like agent programming languages [5, 4]. Finally, it should be noted that besides normative concepts MOISE+ and ISLANDER/AMELI also provide a variety of other social and organizational concepts. Since the focus of this paper is on the normative aspect, the above discussion is limited hereto. Future research will focus on other social and organizational concepts.

3 Programming Multi-Agent Systems with Norms

In this section, we present a programming language to facilitate the implementation of multi-agent systems with norms, i.e., to facilitate the implementation of norm-based artefacts that coordinate/regulate the behavior of participating individual agents. A normative multi-agent system (i.e., a norm-based artefact) is considered to contain two modules: an organization module that specifies norms and sanctions, and an environment module in which individual agents can perform actions. The individual agents are assumed to be implemented in a programming language, not necessarily known to the multi-agent system programmer, though the programmer is required to have the reference to the (executable) programs of each individual agent. It is also assumed that all actions that are performed by individual agents are observable to the multi-agent system (i.e., norm-based artefact). Note that the reference to the (executable) programs of individual agents are required such that multi-agent systems (i.e., normative artefact) can observe the actions generated by the agent programs. Finally, we assume that the effect of an individual agent's action in the external environment is determined by the program that implements the norm-based artefact (i.e., by the multi-agent system program). Most noticeably it is not assumed that the agents are able to reason about the norms of the system.

The programming language for normative multi-agent systems provides programming constructs to specify the effect of an agent's actions in the environment, norms, sanctions, and the initial state of the environment. Moreover, the programming language is based on a monitoring and a sanctioning mechanism that observes the actions performed by the agents, determines their effects in the shared environment, determines the violations caused by performing the actions, and possibly, imposes sanctions. A program in this language is the implementation of a norm-based artefact. As we assume that the norm-based artefacts determine the effects of external actions in the shared environment, the programming language should provide constructs to implement these effects. The effect of an agent's (external) actions is specified by a set of literals that should hold in the shared environment after the external action is performed by the agent. As external actions can have different effects when they are executed in different states of the shared environment, we add a set of literals that function as the pre-condition of those effect.

We consider norms as being represented by counts-as rules [13], which ascribe “institutional facts” (e.g. “a violation has occurred”), to “brute facts” (e.g. “an agent is on the train without ticket”). For example, a counts-as rule may express the norm ”an agent on the train without ticket counts-as a violation”. In our framework, brute facts constitute the environment shared by the agents, while institutional facts constitute the normative/institutional state of the multi-agent system. Institutional facts are used with the explicit aim of triggering system's reactions (e.g., sanctions). As showed in [8] counts-as rules can enjoy a rather classical logical behavior, and are here implemented as simple rules that relate brute and normative facts. In the presented programming language, we distinguish brute facts from normative (institutional) facts and assume two disjoint sets of propositions to denote these facts.

Brute and institutional facts constitute the (initial) state of the multi-agent system (i.e., the state of the norm-based artefact). Brute facts are initially set by the programmer by means of the initial state of the shared environment. These facts can change as individual agents perform actions in the shared environment. Normative facts are determined by applying counts-as rules in multi-agent states. The application of counts-as rules in subsequent states of a multi-agent system realizes a monitoring mechanism as it determines and detects norm violations during the execution of the multi-agent system.

Sanctions are also implemented as rules, but follow the opposite direction of counts-as rules. A sanction rule determines which brute facts will be brought about by the system as a consequence of the normative facts. Typically, such brute facts are sanctions, such as fines. Notice that in human systems sanctions are usually issued by specific agents (e.g. police agents). This is not the case in our computational setting, where sanctions necessarily follow the occurrence of a violation if the relevant sanction rule is in place (comparable to automatic traffic control and issuing tickets). It is important to stress, however, that this is not an intrinsic limitation of our approach. We do not aim at mimicking human institutions but rather providing the specification of computational systems.

3.1 Syntax.

In order to represent brute and institutional facts in our normative multi-agent systems programming language, we introduce two disjoint sets of propositions to denote these facts. The syntax of the normative multi-agent system programming language is presented below using the EBNF notation. In the following, we use `<b-prop>` and `<i-prop>` to be propositional formulae taken from two different disjoint sets of propositions. Moreover, we use `<ident>` to denote a string and `<int>` to denote an integer.

```
N-MAS_Prog   := "Agents: " (<agentName> <agentProg> [<nr>])+ ;
                "Facts: " <bruteFacts>
                "Effects: " <effects>
                "Counts-as rules: " <counts-as>
                "Sanction rules: " <sanctions>;
<agentName>  := <ident>;
<agentProg>  := <ident>;
<nr>         := <int>;
<bruteFacts> := <b-literals>;
<effects>    := ({<b-literals>} <actionName> {<b-literals>})+;
<counts-as>  := ( <literals> ⇒ <i-literals> )+;
<sanctions>  := ( <i-literals> ⇒ <b-literals>)+;
<actionName> := <ident>;
<b-literals> := <b-literal> {"," <b-literal>};
<i-literals> := <i-literal> {"," <i-literal>};
<literals>   := <literal> {"," <literal>};
<literal>    := <b-literal> | <i-literal>;
<b-literal>  := <b-prop> | "not" <b-prop>;
<i-literal>  := <i-prop> | "not" <i-prop>;
```

In order to illustrate the use of this programming language, consider the following underground station example.

```
Agents:           passenger   PassProg   1
Facts:            {-at_platform, -in_train, -ticket}
Effects:          {-at_platform} enter {at_platform},
                  {-ticket} buy_ticket {ticket},
                  {at_platform, -in_train} embark {-at_platform, in_train}
Counts_as rules:  {at_platform , -ticket} ⇒ {viol_1},
                  {in_train , -ticket} ⇒ {viol_⊥}
Sanction rules:   {viol_1} ⇒ {fined_10}
```

This program creates one agent called `passenger` whose (executable) specification is included in a file with the name `PassProg`. The `Facts`, which implement brute facts, determine the initial state of the shared environment. In this case, the agent is not at the platform (`-at_platform`) nor in the train (`-in_train`) and has no ticket (`-ticket`). The `Effects` indicate how the environment can advance in its computation. Each effect is of the form {`pre-condition`} `action` {`post-condition`}. The first effect, for instance, means that if the agent performs an `enter` action when not at the platform, the result is that the agent is on the platform (either with or without a ticket). Only those effects that are changed are thus listed in the post-condition. The `Counts_as rules` determine the normative effects for a given (brute and normative) state of the multi-agent system. The first rule, for example, states that being on the platform without having a ticket is a specific violation (marked by viol_1). The second rule marks states where agents are on a train without a ticket with the specifically designated literal $\mathrm{viol}_\perp$. This literal is used to implement regimentation. The operational semantics of the language ensures that the designated literal $\mathrm{viol}_\perp$ can never hold during any run of the system (see Definition 3). Intuitively, rules with $\mathrm{viol}_\perp$ as consequence could be thought of as placing gates blocking an agent's action. Finally, the aim of `Sanction rules` is to determine the punishments that are imposed as a consequence of violations. In the example the violation of type viol_1 causes the sanction fined_{10} (e.g., a 10 EUR fine).

Counts-as rules obey syntactic constraints. Let $l = (\Phi \Rightarrow \Psi)$ be a rule, we use $\mathtt{cond}_l$ and $\mathtt{cons}_l$ to indicate the condition Φ and consequent Ψ of the rule l, respectively. We consider only sets of rules such that 1) they are finite; 2) they are such that each condition has exactly one associated consequence (i.e., all the consequences of a given conditions are packed in one single set `cons`); and 3) they are such that for counts-as rule k, l, if $\mathtt{cons}_k \cup \mathtt{cons}_l$ is inconsistent (i.e., contains p and $\neg p$), then $\mathtt{cond}_k \cup \mathtt{cond}_l$ is also inconsistent. That is to say, rules trigger inconsistent conclusions only in different states. In the rest of this paper, sets of rules enjoying these three properties are denoted by $\mathbf{R}$.

3.2 Operational Semantics.

One way to define the semantics of this programming language is by means of operational semantics. Using such semantics, one needs to define the configuration (i.e., state) of normative multi-agent systems and the transitions that such configurations can undergo through transition rules. The state of a multiagent system with norms consists of the state of the external environment, the normative state, and the states of individual agents.

Definition 1. *(Normative Multi-Agent System Configuration) Let P_b and P_n be two disjoint sets of literals denoting atomic brute and normative facts (including* $\mathrm{viol}_\perp$*), respectively. Let A_i be the configuration of individual agent i. The configuration of a normative multi-agent system is defined as $\langle \mathcal{A}, \sigma_b, \sigma_n \rangle$ where $\mathcal{A} = \{A_1, \ldots, A_n\}$, σ_b is a consistent set of literals from P_b denoting the brute state of multi-agent system and σ_n is a consistent set of literals from P_n denoting the normative state of multi-agent system.*

The configuration of such a multi-agent system can change for various reasons, e.g., because individual agents perform actions in the external environment or because the external environment can have its own internal dynamics (the state of a clock changes independent of an individual agent's action). In operational semantics, transition rules specify how and when configurations can change, i.e., they specify which transition between configurations are allowed and when they can be derived. In this paper, we consider only the transition rules that specify the transition of multi-agent system configurations as a result of performing external actions by individual agents. Of course, individual agents can perform (internal) actions that modify only their own configurations and have no influence on the multi-agent system configuration. The transition rules to derive such transitions are out of the scope of this paper.

Definition 2. *(Transitions of Individual Agent's Actions) Let A_i and A'_i be configurations of individual agent i, and $\alpha(i)$ be an (observable) external action performed by agent i. Then, the following transition captures the execution of an external action by an agent.*

$$A_i \xrightarrow{\alpha(i)} A'_i \ : \textit{agent } i \textit{ can perform external action } \alpha$$

This transition indicates that an agent configuration can change by performing an external action. The performance of the external action is broadcasted to the multi-agent system level. Note that no assumption is made about the internals of individual agents as we do not present transition rules for deriving internal agent transitions (denoted as $A \longrightarrow A'$). The only assumption is that the action of the agent is observable. This is done by labeling the transition with the external action name.

Before presenting the transition rule specifying the possible transitions of the normative MAS configurations, the closure of a set of conditions under a set of (counts-as and sanction) rules needs to be defined. Given a set $\mathbf{R}$ of rules and a set X of literals, we define the set of applicable rules in X as $\mathtt{Appl}^{\mathbf{R}}(X) = \{\Phi \Rightarrow \Psi \mid X \models \Phi\}$. The closure of X under $\mathbf{R}$, denoted as $\mathtt{Cl}^{\mathbf{R}}(X)$, is inductively defined as follows:

B: $\mathtt{Cl}^{\mathbf{R}}_0(X) = X \cup (\bigcup_{l \in \mathtt{Appl}^{\mathbf{R}}(X)} \mathtt{cons}_l)$
S: $\mathtt{Cl}^{\mathbf{R}}_{n+1}(X) = \mathtt{Cl}^{\mathbf{R}}_n(X) \cup (\bigcup_{l \in \mathtt{Appl}^{\mathbf{R}}(\mathtt{Cl}^{\mathbf{R}}_n(X))} \mathtt{cons}_l)$

Because of the properties of finiteness, consequence uniqueness and consistency of $\mathbf{R}$ one and only one finite number $m+1$ can always be found such that $\mathtt{Cl}^{\mathbf{R}}_{m+1}(X) = \mathtt{Cl}^{\mathbf{R}}_{m}(X)$ and $\mathtt{Cl}^{\mathbf{R}}_{m}(X) \neq \mathtt{Cl}^{\mathbf{R}}_{m-1}(X)$. Let such $m+1$ define the closure X under $\mathbf{R}$: $\mathtt{Cl}^{\mathbf{R}}(X) = \mathtt{Cl}^{\mathbf{R}}_{m+1}(X)$. Note that the closure may become inconsistent due to the ill-defined set of counts-as rules. For example, the counts-as rule $p \Rightarrow -p$ (or the set of counts as rules $\{p \Rightarrow q\ ,\ q \Rightarrow -p\}$), where p and q are normative facts, may cause the normative state of a multi-agent system to become inconsistent.

We can now define a transition rule to derive transitions between normative multi-agent system configurations. In this transition rule, the function *up* determines the effect of action $\alpha(i)$ on the environment σ_b based on its specification $(\Phi\ \alpha(i)\ \Phi')$ as follows:

$$up(\alpha(i), \sigma_b) = (\sigma_b \cup \Phi') \setminus (\{p \mid -p \in \Phi'\} \cup \{-p \mid p \in \Phi'\})$$

Definition 3. *(Transition Rule for Normative Multi-Agent Systems) Let* $\mathbf{R_c}$ *be the set of counts-as rules,* $\mathbf{R_s}$ *be the set of sanction rules, and* $(\Phi\ \alpha(i)\ \Phi')$ *be the specification of action* $\alpha(i)$*. The multi-agent transition rule for the derivation of normative multi-agent system transitions is defined as follows:*

$$\frac{\begin{array}{c} A_i \in \mathcal{A} \quad \& \quad A_i \overset{\alpha(i)}{\rightarrow} A_i' \quad \& \quad \sigma_b \models \Phi \quad \& \quad \sigma_b' = up(\alpha(i), \sigma_b) \\ \sigma_n' = \mathtt{Cl}^{\mathbf{R_c}}(\sigma_b') \setminus \sigma_b' \quad \& \quad \sigma_n' \not\models \mathrm{viol}_\perp \quad \& \quad S = \mathtt{Cl}^{\mathbf{R_s}}(\sigma_n') \setminus \sigma_n' \quad \& \quad \sigma_b' \cup S \not\models \perp \end{array}}{\langle \mathcal{A}, \sigma_b, \sigma_n \rangle \longrightarrow \langle \mathcal{A}', \sigma_b' \cup S, \sigma_n' \rangle}$$

where $\mathcal{A}' = (\mathcal{A} \setminus \{A_i\}) \cup \{A_i'\}$ *and* $\mathrm{viol}_\perp$ *is the designated literal for regimentation.*

This transition rule captures the effects of performing an external action by an individual agent on both external environments and the normative state of the MAS. First, the effect of α on σ_b is computed. Then, the updated environment is used to determine the new normative state of the system by applying all counts-as rules to the new state of the external environments. Finally, possible sanctions are added to the new environment state by applying sanction rules to the new normative state of the system. In should be emphasized that other multi-agent transition rules, such as transition rules for communication actions, are not presented in this paper because the focus here is on how norms determine the effects of external actions.

Note that the external action of an agent can be executed only if it would not result in a state containing $\mathrm{viol}_\perp$. This captures exactly the regimentation of norms. Hence, once assumed that the initial normative state does not include $\mathrm{viol}_\perp$, it is easy to see that the system will never be in a $\mathrm{viol}_\perp$-state. It is important to note that when a normative state σ_n' becomes inconsistent, the proposed transition rule cannot be applied because an inconsistent σ_n' entails $viol_\perp$. Also, note that the condition $\sigma_b' \cup S \not\models \perp$ guarantees that the environment state never can become inconsistent. Finally, it should be emphasized that the normative state σ_b' is not defined on σ_n and is always computed anew.

4 Logic

In this section, we propose a logic to specify and verify liveness and safety properties of multi-agent system programs with norms. This logic, which is a variant of Propositional Dynamic Logic (PDL, see [2]), is in the spirit of [1] and rely on that work. It is important to note that the logic developed in [1] aims at specifying and verifying properties of single agents programmed in terms of beliefs, goals, and plans. Here we modify the logic and apply it to multi-agent system programs. We first introduce some preliminaries before presenting the logic.

4.1 Preliminaries

We show how the programming constructs can be used for grounding a logical semantics. Let P denote the set of propositional variables used to describe brute and normative states of the system. It is assumed that each propositional variable in P denotes either an institutional/normative or a brute state-of-affairs: $P = P_n \cup P_b$ and $P_n \cap P_b = \emptyset$. A state s is represented as a pair $\langle \sigma_b, \sigma_n \rangle$ where $\sigma_b = \{(-)p_1, \ldots, (-)p_n : p_i \in P_b\}$ is a consistent set of literals (i.e., for no $p \in P_b$ it is the case that $p \in \sigma_b$ and $-p \in \sigma_b$), and σ_n is like σ_b for P_n.

Rules are pairs of conditions and consequences $(\{(-)p_1, \ldots, (-)p_n \mid (-)p_i \in X\}, \{(-)q_1, \ldots, (-)q_k \mid (-)q_i \in Y\})$ with X and Y being either σ_b or σ_n when applied in state $\langle \sigma_b, \sigma_n \rangle$. Following [8], if $X = \sigma_b$ and $Y = \sigma_n$ then the rule is called *bridge counts-as rule*; if $X = Y = \sigma_n$ then the rule is an *institutional counts-as rule*; if $X = \sigma_n$ and $Y = \sigma_b$ then the rule is a *sanction rule*. Literals p's and q's are taken to be disjoint. Leaving technicalities aside, bridge counts-as rules connect brute states to normative/institutional ones, institutional counts-as rules connect institutional facts to institutional facts, and sanction rules connect normative states to brute ones.

Given a set $\mathbf{R}$ of rules, we say a state $s = \langle \sigma_b, \sigma_n \rangle$ to be $\mathbf{R}$-aligned if for all pairs $(\mathtt{cond}_k, \mathtt{cons}_k)$ in $\mathbf{R}$: if $\mathtt{cond}_k$ is satisfied either by σ_b (in the case of a bridge counts-as rule) or by σ_n (in the case of an institutional counts-as or a sanction rule), then $\mathtt{cons}_k$ is satisfied by σ_n (in the case of a bridge or institutional counts-as rule) or by σ_b (in the case of a sanction rule), respectively. States that are $\mathbf{R}$-aligned are states which instantiate the normative system specified by $\mathbf{R}$.

Let the set of agents' external actions $\mathtt{Ac}$ be the union $\bigcup_{i \in I} \mathtt{Ac}_i$ of the finite sets $\mathtt{Ac}_i$ of external actions of each agent i in the set I. We denote external actions as $\alpha(i)$ where $\alpha \in \mathtt{Ac}_i$ and $i \in I$. We associate now with each $\alpha(i) \in \mathtt{Ac}_i$ a set of pre- and post-conditions $\{(-)p_1 \in \sigma_b, \ldots, (-)p_n \in \sigma_b\}$, $\{(-)q_1 \in \sigma'_b, \ldots, (-)q_k \in \sigma'_b\}$ (where p's and q's are not necessarily disjoint) when $\alpha(i)$ is executed in a state with brute facts set σ_b which satisfies the pre-condition then the resulting state s' has the brute facts set σ'_b which satisfies the post-condition (including replacing p with $-p$ if necessary to preserve consistency) and it is such that *the rest of* σ'_b *is the same as* σ_b. Executing an action $\alpha(i)$ in different configurations may give different results. For each $\alpha(i)$, we denote the set of pre- and post-condition pairs $\{(\mathtt{prec}_1, \mathtt{post}_1), \ldots, (\mathtt{prec}_m, \mathtt{post}_m)\}$ by $C_b(\alpha(i))$. We assume that $C_b(\alpha(i))$ is finite, that pre-conditions $\mathtt{prec}_k, \mathtt{prec}_l$ are mutually exclusive if $k \neq l$, and that each pre-condition has exactly one associated post-condition. We denote the set of all such pre- and post-conditions of all agents' external actions by $\mathbf{C}$.

Now everything is put into place to show how the execution of $\alpha(i)$ in a state with brute facts set σ_b also univocally changes the normative facts set σ_n by means of the applicable counts-as rules, and adds the resulting sanctions by means of the applicable sanction rules. If $\alpha(i)$ is executed in a state $\langle \sigma_b, \sigma_n \rangle$ with brute facts set σ_b, which satisfies the pre-conditions, then the resulting state $\langle \sigma'_b \cup S, \sigma'_n \rangle$ is such that σ'_b satisfies the brute post-condition of $\alpha(i)$ (including replacing p with $-p$ if necessary) and the rest of σ'_b is the same of σ_b; σ'_n is determined by the closure of σ'_b with counts-as rules $\mathbf{R}_c$; sanctions S are obtained via closure of σ'_n with sanction rules $\mathbf{R}_s$.

4.2 Language

The language L for talking about normative multi-agent system programs is just the language of PDL built out of a finite set of propositional variables $P \cup -P$ (i.e., the literals built from P), used to describe the system's normative and brute states, and a finite set $\texttt{Ac}$ of agents' actions. Program expressions ρ are built out of external actions $\alpha(i)$ as usual, and formulae ϕ of L are closed under boolean connectives and modal operators:

$$\rho ::= \alpha(i) \mid \rho_1 \cup \rho_2 \mid \rho_1 ; \rho_2 \mid ?\phi \mid \rho^*$$
$$\phi ::= (-)p \mid \neg\phi \mid \phi_1 \wedge \phi_2 \mid \langle \rho \rangle \phi$$

with $\alpha(i) \in \texttt{Ac}$ and $(-)p \in P \cup -P$. Connectives $\vee$ and $\rightarrow$, and the modal operator $[\rho]$ are defined as usual.

4.3 Semantics.

The language introduced above is interpreted on transition systems that generalize the operational semantics presented in the earlier section, in that they do not describe a particular program, but all possible programs —according to $\mathbf{C}$— generating transitions between all the $\mathbf{R_c}$ and $\mathbf{R_s}$-aligned states of the system. As a consequence, the class of transition systems we are about to define will need to be parameterized by the sets $\mathbf{C}$, $\mathbf{R_c}$ and $\mathbf{R_s}$.

A model is a structure $M = \langle S, \{R_{\alpha(i)}\}_{\alpha(i) \in \texttt{Ac}}, V \rangle$ where:

- S is a set of $\mathbf{R_c}$ and $\mathbf{R_s}$-aligned states.
- $V = (V_b, V_n)$ is the evaluation function consisting of brute and normative valuation functions V_b and V_n such that for $s = \langle \sigma_b, \sigma_n \rangle$, $V_b(s) = \sigma_b$ and $V_n(s) = \sigma_n$.
- $R_{\alpha(i)}$, for each $\alpha(i) \in \texttt{Ac}$, is a relation on S such that $(s, s') \in R_{\alpha(i)}$ iff for some $(\texttt{prec}_k, \texttt{post}_k) \in C(\alpha(i))$, $\texttt{prec}_k(s)$ and $\texttt{post}_k(s')$, i.e., for some pair of pre- and post-conditions of $\alpha(i)$, the pre-condition holds for s and the corresponding post-condition holds for s'. Note that this implies two things.

First, an $\alpha(i)$ transition can only originate in a state s which satisfies one of the pre-conditions for $\alpha(i)$. Second, since pre-conditions are mutually exclusive, every such s satisfies exactly one pre-condition, and all $\alpha(i)$-successors of s satisfy the matching post-condition.

Given the relations corresponding to agents' external actions in M, we can define sets of paths in the model corresponding to any PDL program expression ρ in M. A set of paths $\tau(\rho) \subseteq (S \times S)^*$ is defined inductively:

- $\tau(\alpha(i)) = \{(s, s') : R_{\alpha(i)}(s, s')\}$
- $\tau(\phi?) = \{(s, s) : M, s \models \phi\}$
- $\tau(\rho_1 \cup \rho_2) = \{z : z \in \tau(\rho_1) \cup \tau(\rho_2)\}$
- $\tau(\rho_1; \rho_2) = \{z_1 \circ z_2 : z_1 \in \tau(\rho_1),\ z_2 \in \tau(\rho_2)\}$, where $\circ$ is concatenation of paths , such that $z_1 \circ z_2$ is only defined if z_1 ends in the state where z_2 starts
- $\tau(\rho^*)$ is the set of all paths consisting of zero or finitely many concatenations of paths in $\tau(\rho)$ (same condition on concatenation as above)

Constructs such as `If` ϕ `then` ρ_1 `else` ρ_2 and `while` ϕ `do` ρ are defined as $(\phi?; \rho_1) \cup (\neg\phi?; \rho_2)$ and $(\phi?; \rho)^*; \neg\phi$, respectively. The satisfaction relation $\models$ is inductively defined as follows:

- $M, s \models (-)p$ iff $(-)p \in V_b(s)$ for $p \in P_b$
- $M, s \models (-)p$ iff $(-)p \in V_n(s)$ for $p \in P_n$
- $M, s \models \neg\phi$ iff $M, s \not\models \phi$
- $M, s \models \phi \wedge \psi$ iff $M, s \models \phi$ and $M, s \models \psi$
- $M, s \models \langle\rho\rangle\phi$ iff there is a path in $\tau(\rho)$ starting in s which ends in a state s' such that $M, s' \models \phi$.
- $M, s \models [\rho]\phi$ iff for all paths $\tau(\rho)$ starting in s, the end state s' of the path satisfies $M, s' \models \phi$.

Let the class of transition systems defined above be denoted $\mathbf{M}_{\mathbf{C},\mathbf{R_c},\mathbf{R_s}}$ where $\mathbf{C}$ is the set of pre- and post-conditions of external actions, $\mathbf{R_c}$ is the set of counts-as rules and $\mathbf{R_s}$ the set of sanction rules.

4.4 Axiomatics.

The axiomatics shows in what the logic presented differs w.r.t. standard PDL. In fact, it is a conservative extension of PDL with domain-specific axioms needed to axiomatize the behavior of normative multi-agent system programs.

For every pre- and post-condition pair $(\mathtt{prec}_i, \mathtt{post}_i)$ we describe states satisfying $\mathtt{prec}_i$ and states satisfying $\mathtt{post}_i$ by formulas of L. More formally, we define a formula $\mathit{tr}(X)$ corresponding to a pre- or post-condition X as follows: $\mathit{tr}((-)p) = (-)p$ and $\mathit{tr}(\{\phi_1, \ldots, \phi_n\}) = \mathit{tr}(\phi_1) \wedge \ldots \wedge \mathit{tr}(\phi_n)$. This allows us to axiomatize pre- and post-conditions of actions. The conditions and consequences of counts-as rules and sanction rules can be defined in similar way as pre- and post-conditions of actions, respectively. The set of models $\mathbf{M}_{\mathbf{C},\mathbf{R_c},\mathbf{R_s}}$ is axiomatized as follows:

PDL Axioms and rules of PDL

Ax Consistency Consistency of literals: $\neg(p \wedge -p)$

Ax Counts-as For every rule $(\texttt{cond}_k, \texttt{cons}_k)$ in $\mathbf{R_c}$: $\mathit{tr}(\texttt{cond}_k) \rightarrow \mathit{tr}(\texttt{cons}_k)$

Ax Sanction For every rule $(\texttt{viol}_k, \texttt{sanc}_k)$ in $\mathbf{R_s}$: $\mathit{tr}(\texttt{viol}_k) \rightarrow \mathit{tr}(\texttt{sanc}_k)$

Ax Regiment $\text{viol}_\perp \rightarrow \perp$

Ax Frame For every action $\alpha(i)$ and every pair of pre- and post-conditions $(\texttt{prec}_j, \texttt{post}_j)$ in $C(\alpha(i))$ and formula Φ built out of P_b not containing any propositional variables occurring in $\texttt{post}_j$:

$$\mathit{tr}(\texttt{prec}_j) \wedge \Phi \rightarrow [\alpha(i)](\mathit{tr}(\texttt{post}_\texttt{j}) \wedge \Phi)$$

This is a frame axiom for actions.

Ax Non-Executability For every action $\alpha(i)$, where all possible pre-conditions in $C(\alpha(i))$ are $\texttt{prec}_1, \ldots, \texttt{prec}_k$: $\neg\mathit{tr}(\texttt{prec}_1) \wedge \ldots \wedge \neg\mathit{tr}(\texttt{prec}_k) \rightarrow \neg\langle\alpha(i)\rangle\top$ where $\top$ is a tautology.

Ax Executability For every action $\alpha(i)$ and every pre-condition $\texttt{prec}_j$ in $C(\alpha(i))$: $\mathit{tr}(\texttt{prec}_j) \rightarrow \langle\alpha(i)\rangle\top$

Let us call the axiom system above $\mathbf{Ax_{C,R_c,R_s}}$, where $\mathbf{C}$ is the set of brute pre- and post-conditions of atomic actions, $\mathbf{R_c}$ is the set of counts-as rules, and $\mathbf{R_s}$ is the set of sanction rules.

Theorem 1. *Axiomatics* $\mathbf{Ax_{C,R_c,R_s}}$ *is sound and weakly complete for the class of models* $\mathbf{M_{C,R_c,R_s}}$.

Proof. Soundness is proven as usual by induction on the length of derivations. We sketch the proof of completeness. It builds on the usual completeness proof of PDL via finite canonical models. Given a consistent formula ϕ to be proven satisfiable, such models are obtained via the Fischer-Ladner closure of the set of subformulae of the formula ϕ extended with all pre- and post-conditions of any action $\alpha(i)$ occurring in ϕ. Let $FLC(\phi)$ denote such closure. The canonical model consists of all the maximal $\mathbf{Ax_{C,R_c,R_s}}$-consistent subsets of $FLC(\phi)$. The accessibility relation and the valuation of the canonical model are defined like in PDL and the truth lemma follows in the standard way. It remains to be proven that the model satisfies the axioms. First, since the states in the model are maximal and consistent w.r.t. *Ax Counts-as*, *Ax Sanction*, *Ax Consistency*, and *AxRegiment*, they are $\mathbf{R_c}$- and $\mathbf{R_s}$-aligned, σ_b and σ_n are consistent, and no state is such that $\sigma_n \models \texttt{viol}_\perp$. Second, it should be shown that the canonical model satisfies the pre- and post-conditions of the actions occurring in ϕ in that: a) no action $\alpha(i)$ is executable in a state s if none of its preconditions are satisfied by s, and b) if they hold in s then the corresponding post-conditions hold in s' which is accessible by$R_{\alpha(i)}$ from s. As to a), if a state s in the canonical model does not satisfy any of the preconditions of $\alpha(i)$ then, by *Ax Non-Executability* and the definition of the canonical accessibility relation, there is no s' in the model such that $sR_{\alpha(i)}s'$. As to b), if a state s in the canonical model satisfies one of the preconditions $\texttt{prec}_\texttt{j}$ of $\alpha(i)$ then$\mathit{tr}(\texttt{prec}_\texttt{j})$ belongs to s and, by *Ax Frame*, $[\alpha(i)]\mathit{tr}(\texttt{post}_\texttt{j})$ also do. Now, *Ax Executability* guarantees that there exists at least one s' such that $sR_{\alpha(i)}s'$, and, for any s' such that $sR_{\alpha(i)}s'$, by the definition of such canonical accessibility relation, s' contains $\mathit{tr}(\texttt{post}_\texttt{j})$ (otherwise it would not be the case that $sR_{\alpha(i)}s'$). On the other hand, for any literal $(-)p$ in s not occurring in $tr(\texttt{post}_\texttt{j})$, its value cannot change from s to s' since, if it would, then for *Ax Frame* it would not be the case that $sR_{\alpha(i)}s'$, which is impossible. This concludes the proof.

4.5 Verification

To verify a normative multi-agent system program means, in our perspective, to check whether the program implementing the normative artefact is soundly designed w.r.t. the regimentation and sanctioning mechanisms it is supposed to realize or, to put it in more general terms, to check whether certain property holds in all (or some) states reachable by the execution traces of the multi-agent system program. In order to do this, we need to translate a multi-agent system program into a PDL program expression.

As explained in earlier sections, a multi-agent system program assumes a set of behaviors $A_1, \ldots, A_n$ of agents $1, \ldots, n$, each of which is a sequence of external actions (the agents actions observed from the multi-agent level), i.e., $A_i = \alpha_i^1; \alpha_i^2, \ldots$ where $\alpha_i^j \in Ac$. [1] Moreover, a multi-agent system program with norms consists of an initial set of brute facts, a set of counts-as rules and a set of sanction rules which together determine the initial state of the program. In this paper, we consider the execution of a multi-agent program as interleaved executions of the involved agents' behaviors started at the initial state.

Given I as the set of agents' names and A_i as the behavior of agent $i \in I$, the execution of a multi-agent program can be described as PDL expression $\bigcup$ *interleaved*$(\{A_i | i \in I\})$, where *interleaved*$(\{A_i | i \in I\})$ yields all possible interleavings of agents' behaviors, i.e., all possible interleavings of actions from sequences A_i. It is important to notice that $\bigcup$ *interleaved*$(\{A_i | i \in I\})$ corresponds to the set of computations sequences (execution traces) generated by the operational semantics.

The general verification problem can now be formulated as follows. Given a multi-agent system program with norms in a given initial state satisfying $\phi \in L$, the state reached after the execution of the program satisfies ψ, i.e.:

$$\phi \rightarrow \langle [\bigcup \mathit{interleaved}(\{A_i | i \in I\})] \rangle \psi$$

In the above formulation, the modality $\langle [\ldots] \rangle$ is used to present both safety $[\ldots]$ and liveness $\langle \ldots \rangle$ properties. We briefly sketch a sample of such properties using again the multi-agent system program with norms which implements the train station example with one passenger agent (see Section 3).

Sanction follows violation. Entering without a ticket results in a fine, i.e.,

$$\neg\texttt{at\_platform} \wedge \neg\texttt{train} \wedge \neg\texttt{ticket} \rightarrow [\texttt{enter}](\texttt{viol}_1 \wedge \texttt{pay}_{10}).$$

[1] Note an agent's behavior can always be written as a (set of) sequence(s) of actions, which in turn can be written as a PDL expressions.

Norm obedience avoids sanction. Buying a ticket if you have none and entering the platform does not result in a fine, i.e.:

$$-\texttt{at\_platform}\wedge-\texttt{train} \rightarrow \langle\,\text{If}-\texttt{ticket}\,\text{then}\,\texttt{buy\_ticket};\texttt{enter}\,\rangle\,(\texttt{at\_platform}\wedge-\texttt{pay}_{10}).$$

Regimentation. It is not possible for an agent to enter the platform and embark the train without a ticket, i.e.:

$$-\texttt{at\_platform}\wedge-\texttt{train}\wedge-\texttt{ticket} \rightarrow [\texttt{enter};\texttt{embark}]\bot$$

Note that there is only one passenger agent involved in the example program. For this property, we assume that the passenger's behavior is $\texttt{enter};\texttt{embark}$. Note also that:

$$\bigcup \mathit{interleaved}(\{\texttt{enter};\texttt{embark}\}) = \texttt{enter};\texttt{embark}.$$

Below is the proof of the regimentation property above with respect to the multi-agent system program with norms that implements the train station with one passenger.

Proof. First, axiom *Ax Frame* using the specification of the *enter* action (with pre-condition $\{\texttt{-at\_platform}\}$ and post-condition $\{\texttt{at\_platform}\}$) gives us
(1) $-\texttt{at\_platform}\wedge-\texttt{in\_train}\wedge-\texttt{ticket} \rightarrow$
$[\texttt{enter}]\ \texttt{at\_platform}\wedge-\texttt{in\_train}\wedge-\texttt{ticket}$
Moreover, axiom *Ax Frame* using the specification of the *embark* action (with pre-condition $\{\texttt{at\_platform, -in\_train}\}$ and post-condition $\{\texttt{-at\_platform, in\_train}\}$) gives us
(2) $\texttt{at\_platform}\wedge-\texttt{in\_train}\wedge-\texttt{ticket} \rightarrow$
$[\texttt{embark}]\ -\texttt{at\_platform}\wedge\texttt{in\_train}\wedge-\texttt{ticket}$
Also, axiom *Ax Counts-as* and the specification of the second counts-as rule of the program give us
(3) $\texttt{in\_train}\wedge-\texttt{ticket}\rightarrow\texttt{viol}_{\bot}$
And axiom *Ax Regiment* together with formula (3) gives us
(4) $\texttt{in\_train}\wedge-\texttt{ticket}\rightarrow\bot$
Now, using PDL axioms together with formula (1), (2), and (4) we get first
(5) $-\texttt{at\_platform}\wedge-\texttt{in\_train}\wedge-\texttt{ticket} \rightarrow [\texttt{enter}][\texttt{embark}]\ \bot$
and thus
(6) $-\texttt{at\_platform}\wedge-\texttt{in\_train}\wedge-\texttt{ticket} \rightarrow [\texttt{enter};\texttt{embark}]\ \bot$. This completes the derivation.

5 Conclusions and Future Work

The paper has proposed a programming language for implementing multi-agent systems with norms. The programming language has been endowed with formal operational semantics, therefore formally grounding the use of certain social notions —eminently the notion of norm, regimentation and enforcement— as explicit programming constructs. A sound and complete logic has then been proposed which can be used for verifying properties of the multi-agent systems with norms implemented in the proposed programming language.

We have already implemented an interpreter for the programming language that facilitates the implementation of multi-agent systems without norms (see `http://www.cs.uu.nl/2apl/`). Currently, we are working to build an interpreter for the modified programming language. This interpreter can be used to execute programs that implement multi-agent systems with norms. Also, we are working on using the presented logic to devise a semi-automatic proof checker for verification properties of normative multi-agent programs.

We are aware that for a comprehensive treatment of normative multi-agent systems we need to extend our framework in many different ways. Future work aims at extending the programming language with constructs to support the implementation of a broader set of social concepts and structures (e.g., roles, power structure, task delegation, and information flow), and more complex forms of enforcement (e.g., policing agents) and norm types (e.g., norms with deadlines). Another extension of the work is the incorporation of the norm-awareness of agents in the design of the multi-agent system. We also aim at extending the framework to capture the role of norms and sanctions concerning the interaction between individual agents.

The approach in its present form concerns only closed multi-agent systems. Future work will also aim at relaxing this assumption providing similar formal semantics for open multi-agent systems. Finally, we have focused on the so-called 'ought-to-be' norms which pertain to socially preferable states. We intend to extend our programming framework with 'ought-to-do' norms pertaining to socially preferable actions.

References

1. N. Alechina, M. Dastani, B. Logan, and J.-J.Ch Meyer. A logic of agent programs. In *Proc. AAAI 2007*, 2007.
2. P. Blackburn, M. de Rijke, and Y. Venema. *Modal Logic.* Cambridge University Press, Cambridge, 2001.
3. C. Castelfranchi. Formalizing the informal?: Dynamic social order, bottom-up social control, and spontaneous normative relations. *JAL*, 1(1-2):47–92, 2004.
4. M. Dastani. 2apl: a practical agent programming language. *International Journal of Autonomous Agents and Multi-Agent Systems*, 16(3):214–248, 2008.
5. M. Dastani and J.-J. Meyer. A practical agent programming language. In *In Proc. of ProMAS'07*, 2008.
6. V. Dignum. *A Model for Organizational Interaction.* PhD thesis, Utrecht University, SIKS, 2003.
7. M. Esteva, J.A. Rodríguez-Aguilar, B. Rosell, and J.L. Arcos. Ameli: An agent-based middleware for electronic institutions. In *Proc. of AAMAS 2004*, New York, US, July 2004.
8. D. Grossi. *Designing Invisible Handcuffs.* PhD thesis, Utrecht University, SIKS, 2007.

9. J. F. Hübner, J. S. Sichman, and O. Boissier. Moise+: Towards a structural functional and deontic model for mas organization. In *Proc. of AAMAS 2002*. ACM, July 2002.
10. A. J. I. Jones and M. Sergot. On the characterization of law and computer systems. In *Deontic Logic in Computer Science*. 1993.
11. Rosine Kitio, Olivier Boissier, Jomi Fred Hbner, and Alessandro Ricci. Organisational artifacts and agents for open multi-agent organisations: giving the power back to the agents. In *Coordination, Organizations, Institutions, and Norms in Agent Systems III*, volume 4870, pages 171–186. Springer, 2007.
12. A. Ricci, M. Viroli, and A. Omicini. "Give agents their artifacts": The A&A approach for engineering working environments in MAS. In *In proc. of AAMAS 2007*, Honolulu, Hawai'i, USA, 2007.
13. J. Searle. *The Construction of Social Reality*. Free, 1995.
14. F. Zambonelli, N. Jennings, and M. Wooldridge. Developing multiagent systems: the GAIA methodology. *ACM Transactions on Software Engineering and Methodology*, 12(3):317–370, 2003.

On dissemination mechanism of corporate social responsibility (CSR): Analysis with agent simulation

Takashi Hashimoto[1], Naoto Shinohara[2], and Susumu Egashira[3]

[1] School of Knowledge Science, Japan Advanced Institute of Science and Technology
1-1, Asahidai, Nomi, Ishikawa, 923-1292, Japan
hash@jaist.ac.jp
[2] School of Knowledge Science, Japan Advanced Institute of Science and Technology
1-1, Asahidai, Nomi, Ishikawa, 923-1292, Japan
shinohara@jaist.ac.jp
[3] Otaru University of Commerce
3-5-21, Midori, Otaru, 047-8501, Japan
egashira@res.otaru-uc.ac.jp

Abstract. Corporate Social Responsibility (CSR), such as pro-environmental behaviour and fair trade, is a kind of normative behaviour by private companies to provide a quasi-public good. We study dissemination mechanism of CSR with a multi-agent model in which corporation agents and consumer agents interact with each other. We show that the mechanism to disseminate CSR is a positive feedback between the corporations' popularity seeking behaviour and the consumers' social learning in which CSR-seeking preference is evaluated according to both the local average of the preferences of surrounding consumers and the global average of the investment in CSR by all corporations. We also discuss an institutional design to establish CSR from an objectionable social state.

Keywords. CSR (corporate social responsibility), Quasi-public good, Institutional design, Positive Feedback, Multi-agent simulation

1 Introduction

Corporate Social Responsibility (CSR) is to take responsibility by organizations including private companies for the effect of their activities on all stakeholders such as customers, employees, shareholders, investors, communities and so on. A typical CSR activity is pro-environmental behaviour. CSR is a kind of normative action by private companies and is considered as an enlightened movement, since such activity must contribute to sustainability. From economic theoretical viewpoint, however, it is said that CSR is inefficient and impossible to maintain in a competitive market. If we consider CSR is an effective means for attaining sustainable society, we need to know a scheme to disseminate and to establish CSR activities in our society. Further, considering the scheme, both theoretically

Dagstuhl Seminar Proceedings 09121
Normative Multi-Agent Systems
http://drops.dagstuhl.de/opus/volltexte/2009/1911

and empirically, may lead us to an understanding of a method to make agents behave normatively.

In this paper, we study dissemination mechanism of CSR activities using a multi-agent simulation. We think of CSR as an institution. Here we use the term institution in Veblen's sense, namely an institution is a habit of thought common to the majority of individuals in a society [1]. A habit of thought is a kind of a value or a preference. In order to establish CSR activities in a society, stakeholders must have value/preference that CSR activities are nice or agreeable for them in spite of demerit such as higher prices of private goods. Although preference of people must change in order to disseminate CSR, preference is taken as fixed in the standard economic theory. Thus, we consider a multi-agent system for flexible modelling of agents' thought and behaviour.

2 Modelling CSR

2.1 Conceptualization of CSR

Before making a multi-agent model of CSR, we need to conceptualise CSR itself. Here we look at CSR as to provide both a private good and a quasi-public good by companies in their activities. A quasi-public good means the following two.

- Consuming the good by any individuals is not excluded (non-excludability) and one's consumption lessens the benefit of others' consumption (rivalness). Global environment is an example.
- A good produces private benefit in addition to social benefit. Donation is an example, in which we can feel satisfaction which is a private benefit.

2.2 Multi-agent modelling of CSR dissemination

We describe an outline of our multi-agent model. In the model, there are two types of agents, corporations and consumers, both are aligned in two-dimensional planes, respectively. The two-dimensional planes are models of relational space of the agents, not physical/geographical spaces.

The corporations produce private and quasi-public goods simultaneously. They have strategies to decide ratios of invest in the quasi-public goods, $\theta_i = [0, 1]$, called "invest ratio". The higher the invest ratio of a corporation is, the higher the price of the corporation's product is. The dissemination of CSR in this model is defined as follows: most corporations are going to take very high value of the invest ratio.

Each consumer buys a product of a corporation. The consumers have preferences that decide the ratio of importance of the quasi-public good to the price of the private good, denoted by $\alpha_j = [0, 1]$ and called "importance ratio". Each consumer chooses one corporation according to his/her importance ratio, where he/she is likely to adopt a corporation having the similar value of the invest ratio to that of his/her importance ratio. This choice is not always exact. We use the Boltzmann distribution in order to take probabilistic choice into account.

The corporations change their strategies according to the popularity share, that is, how they are adopted by the consumers. They try to copy the invest ratio of the most popular corporation in their vicinity (8 agents). This adoption is also not exact. A normal random noise with 0 mean and 0.1 standard deviation is added to the original value.

Concerning the preference of consumers, we investigate four models, one is fixed and three adaptive:

1. fixed preference
2. income standard
3. local standard
4. glocal standard (glocal means global + local)

Each consumer evaluates his/herself and surrounding consumers (8 agents) according to a standard, which is different point among the three adaptive preference models.

In the second model, the consumers refer to the income as the standard. Since a consumer with higher value of the importance ratio buys a product with higher price, his/her income gets lower. Therefore, the income standard is the synonym for disregard of the quasi-public good.

In the third model, the local standard, the consumer uses a weighted arithmetic average of the disregard, $(1-\alpha)$, and the importance ratio, α, of the quasi-public good to evaluate themselves. The weight of the importance ratio is the local average of the importance ratio of the neighbouring agents. The evaluation standard is defined by the following equation:

$$V_{ij}^{\mathrm{L}}(t) = (1 - \langle \alpha_i(t) \rangle)(1 - \alpha_{ij}(t)) + \langle \alpha_i(t) \rangle \alpha_{ij}(t) \ , \tag{1}$$

where

- $V_{ij}^{\mathrm{L}}(t)$: the evaluation of the jth consumer of the ith consumer's neighbour at the tth period in the local standard model,
- $\langle \alpha_i(t) \rangle$: the average of α of 8 consumers in the neighbour of the ith consumer at the tth period,
- $\alpha_{ij}(t)$: the importance ratio of the jth consumer of the ith consumer's neighbour at the tth period.

In the fourth model, the glocal standard, the consumers use both global and local information to evaluate themselves. While the local information is the same as the local standard model, $\langle \alpha_i(t) \rangle$, the global information is the average of invest ratios, θs of all the corporations' and is used only for the weight of the importance ratio term. That is,

$$V_{ij}^{\mathrm{GL}}(t) = (1 - \langle \alpha_i(t) \rangle)(1 - \alpha_{ij}(t)) + (\langle \alpha_i(t) \rangle + \langle \theta(t-1) \rangle)\, \alpha_{ij}(t) \ , \tag{2}$$

where

- $V_{ij}^{\mathrm{GL}}(t)$: the evaluation of the jth consumer of the ith consumer's neighbour at the tth period in the glocal standard model,

- $\langle\theta(t-1)\rangle$: the average of invest ratio θ of all corporations at the $(t-1)$th period.

We use social learning, i.e., imitation, for the adaptive change of agents' preferences. The reason why we take social learning is that we consider agents as social individuals that the individuals' ways of thought, including values, preferences and cognitive frameworks, develop through interactions with others in a society[2]. In each adaptive model, if a consumer takes the lowest in the neighbour according to the standard, he/she imitates the importance ratio of the highest consumer in the surrounding. The imitated value is perturbed by a normal random number with 0 mean and 0.1 standard deviation.

3 Simulation Results

We summarise the results of computer simulations of the above four models. The sizes of the corporation plane and the consumer plane are 10×10 and 100×100, respectively. Thus, the total number of corporations and consumers are 100 and 10000, respectively. The initial states of the invest ratios and the importance ratios are prepared with a uniform distribution.

There is no interesting phenomenon in the fixed preference and the income standard models. In the fixed preference model, the CSR is not disseminated. A little corporations takes high invest ratio. But that is a mere reflection of the existence of the consumers with high importance ratio prepared by the initial uniform distribution. This result is qualitatively equivalent to an economic theoretical model [3]. This is a reasonable consequence since the fixed preference is the equivalent setting to the presumption of the standard economic theory.

Since the income standard means disregard of the quasi-public good, as we mentioned already, the society is occupied by the consumers with very low importance ratio in the income standard model. As they select cheap price products, the corporations also take the low invest strategy in the quasi-public good. The CSR fades away.

3.1 Local Standard Model

In the local standard model, the frequencies of the corporations' invest ratios and that of the consumers' importance ratios change with time as shown in Fig. 1. These graphs represent the dynamics of histograms with 0.1 bin width. The consumers converge to a distribution in which both large ($\alpha \geq 0.8$) and small ($\alpha \leq 0.2$) importance ratios have greater volumes. Other importance ratios are almost even. On the other hand, the corporations continually change their invest ratios. They pursue popularity share by changing their strategies. If a corporation has a top popularity, surrounding corporations come to take the similar invest ratio to the top corporation. As the number of corporations increase at the ratio range of the top popularity, they share the consumers. As a consequence, the popularity of each corporation at this ratio range declines.

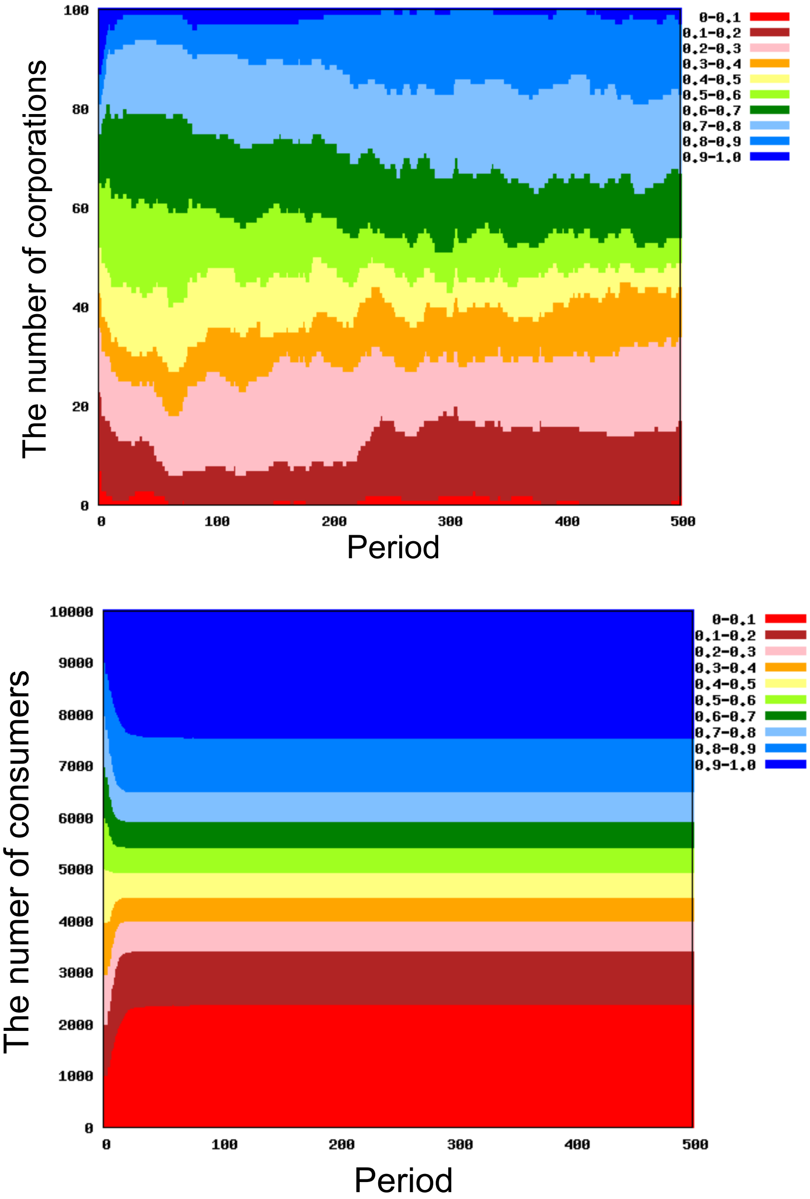


Fig. 1. The dynamics of the frequency of the corporations' invest ratios (upper) and that of the consumers' importance ratios (lower) in the local standard model. The x axis is the period and the y axis is the number of agents. The color legend is shown at the right upper corner.

By keeping such popularity seeking action, the frequency of the invest ratio does not converge to a fixed state.

Figure 2 shows the spatial configuration of the consumers' importance ratio in their two-dimensional plane. Interestingly, the plane converge to an inhomogeneous state. The consumers with high importance ratio (blue) and those with low ratio (red) form clusters. The intermediate ratio consumers are at the boundaries of two types of clusters.

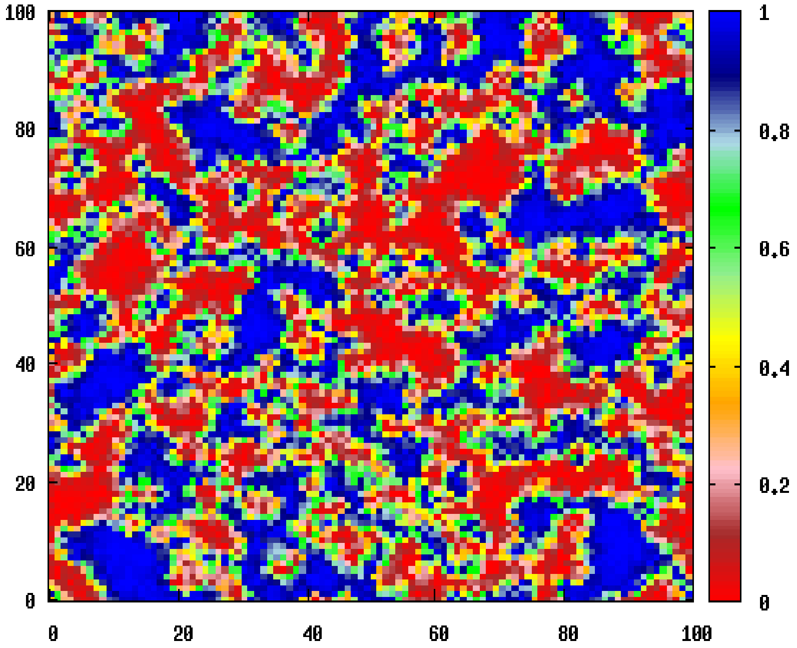


Fig. 2. The spatial distribution of the consumers' importance ratios in the local standard model at the 500th period. The color legend is shown at the right of the figure.

The results shown here are similar to the reality. An empirical investigation says that consciousness about CSR and environment-friendliness is stratified in a society [4]. There are individuals with high consciousness followed by middle and low. People with high consciousness often form groups. People inside of the group have strong relationships.

The spatial inhomogeneity remains because the consumer agents in the local standard model refer local (their surrounding) consumers to evaluate themselves, that is $\langle \alpha_i(t) \rangle$. There is no absolute standard. Under this circumstance, such a situation occurs that an agent A is the lowest rank according to an agent B's standard, but the agent B is the lowest rank according to the agent A's standard. In this situation, both agents do not have incentives to change themselves, thus the inhomogeneity and diversity are maintained.

Although the observed phenomena is realistic as well as interesting, and corporations with high invest ratio exist to some extent, the ratio of them is not enough. We conclude that CSR is not disseminated in the local standard model.

3.2 Glocal Standard Model

The change of the frequency of the corporations' invest ratios and that of the consumers' importance ratios are depicted in Fig. 3. In this case, very large importance ratio ($\alpha \geq 0.9$) permeates rapidly among the consumers. Behind the consumers movement, corporations with very large invest ratio ($\theta \geq 0.9$) also increase and finally occupy the society. We consider this state as CSR established. The spatial distribution of the consumers' importance ratios has islands in which consumers having very low importance ratio are at the core (Fig. 4).

We introduce the average of invest ratio of all corporation as global information into the weight for the second term of the evaluation standard (refer to eq. (2)) that puts importance on CSR, but not into that of disregarding CSR (first term of eq. (2)). This is a trick to expand large invest ratios. The mechanism to disseminate CSR is the followings:

1. There are consumers with high importance ratios to some extent.
2. Such consumers choose corporations with high invest ratio.
3. In order to improve *popularity*, corporations imitate the strategies of the chosen corporations.
4. As a result, the average of the corporations' invest ratio, $\langle \theta(t) \rangle$, increases.
5. The weight of the CSR-seeking term in the glocal evaluation standard, refer to eq. (2), increases and the *social learning* by the consumers is directed to higher importance ratio.
6. The consumers are likely to increase their importance ratio. (back to 2.)

This is a positive feedback mechanism between the corporations' popularity seeking behaviour and the consumers' social learning. In this feedback loop, the consumers' CSR-seeking preference is evaluated according to both the local average of the preferences and the global average of all corporations' investment in CSR, and then the consumers learn socially, i.e., imitate locally. As a phenomenon level, this positive feedback is also observed as a mutual strengthen between the consumers' consciousness putting importance on CSR and the corporates' investment in the quasi-public good. Here, the important point is that

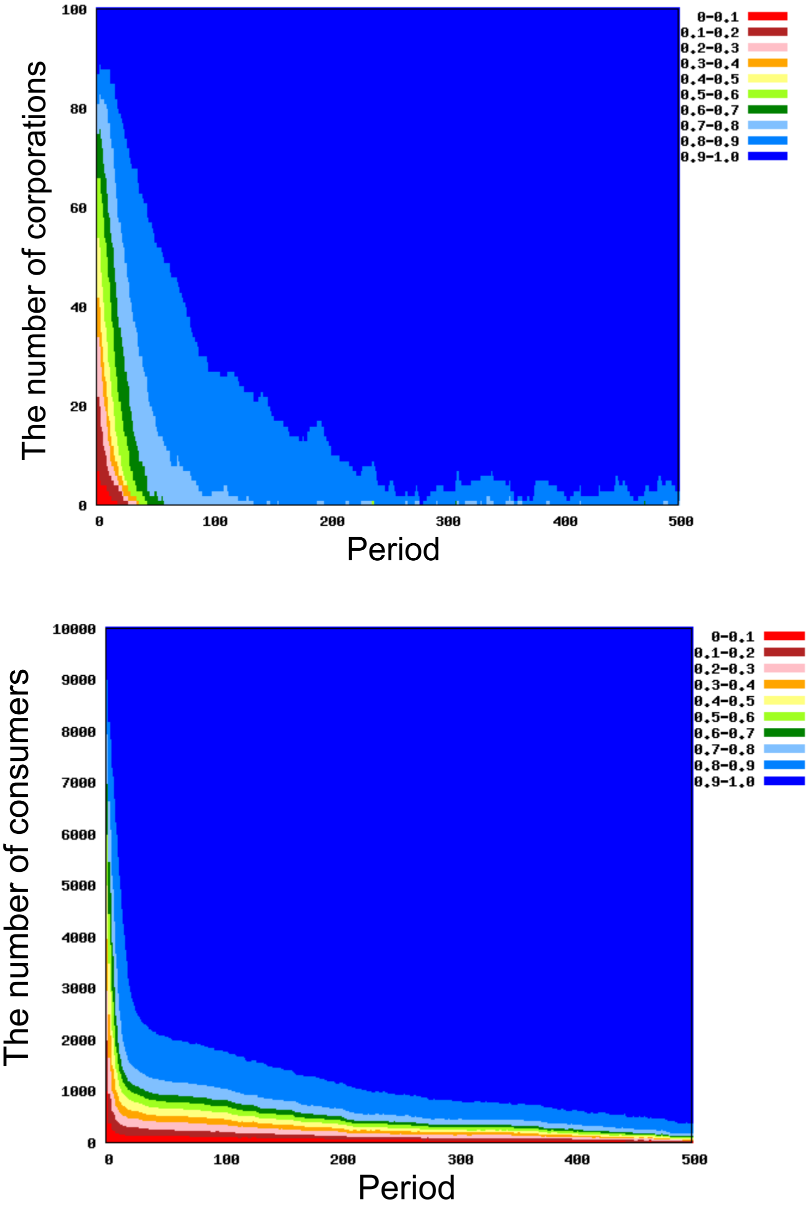


Fig. 3. The dynamics of the frequency of the corporations' invest ratios (upper) and that of the consumers' importance ratios (lower) in the glocal standard model. The x axis is the period and the y axis is the number of agents. The color legend is shown at the right upper corner.

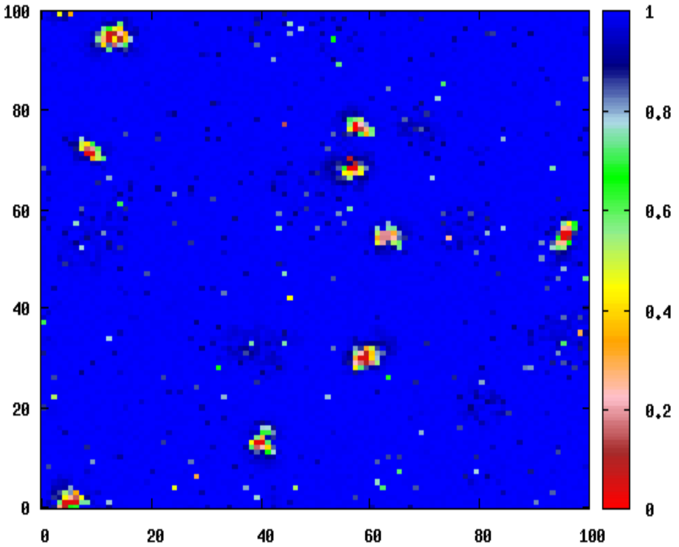


Fig. 4. The spatial distribution of the consumers' importance ratios in the glocal standard model. The color legend is shown at the right of the figure.

the corporations take just myopic actions, popularity seeking. The spread of the importance of CSR in the consumers precedes the strategy of corporations. That is, an institution to putting importance on CSR establishes in the sub-parts of consumer society first. The corporations pursue the institution, and this action promotes an atmosphere or a public opinion that CSR is important. The atmosphere, then, boosts the consumers' change to higher importance ratio through social learning.

4 Institutional Design

If the positive feedback mechanism works well, even in the society under an objectionable state, that is, only with low importance ratio consumers, CSR may be disseminated. We are able to also consider an institutional design in which a policy is employed to realise a desirable social state. Here we suppose a policy to regulate all corporations to maintain the least ratio of the investment in the quasi-public good.

We investigate how CSR is established by controlling the minimum value of the invest ratio of the corporations and the maximum value of the importance ratio of the consumers at the initial state. The former corresponds to the strength of the regulation by the policy and the latter the desirability of social state. We calculate the average of the consumers' importance ratio at the final (converged) state. While the definition of the establishing CSR in this paper is maintaining the higher invest ratio in a quasi-public good by the corporations, we observe the state of the consumer society, since we already know that establishing the institutions in the consumers leads the dissemination of CSR.

In Fig.5, the result of this calculation is shown. We can see two regions, establishment of CSR (the value of z is 1.0) and complete loss of CSR (the value of z is 0.0). There is a steep cliff between these two regions. This is because there is a threshold for the positive feedback mechanism above mentioned to work. This result suggests that if the social state is not so desirable but not too objectionable, say the maximum of α is 0.6, the regulation by the policy need not to be so strong, the minimum θ is 0.3.

In order to disseminate and establish CSR, we need to constitute a society like the fourth model, the glocal standard model. But the reality at the present seems to be in the third model, the local standard. The difference is whether people take the global information such as an atmosphere putting importance on CSR into consideration to evaluate themselves. From the viewpoint of institutional design, a possible way for the shift from the local to the glocal standard is to give publicity to the present status of corporations' CSR activities by municipality or government. When we successfully shift to the society supposed by the glocal standard model, there are two scenarios. One is an optimistic scenario in which CSR disseminate and is established by itself if there are consumers with enough high consciousness about CSR. The other is less optimistic one, we need a regulation on corporations activities to make a least level of investment in a quasi-public good, if there is not enough high conscious people.

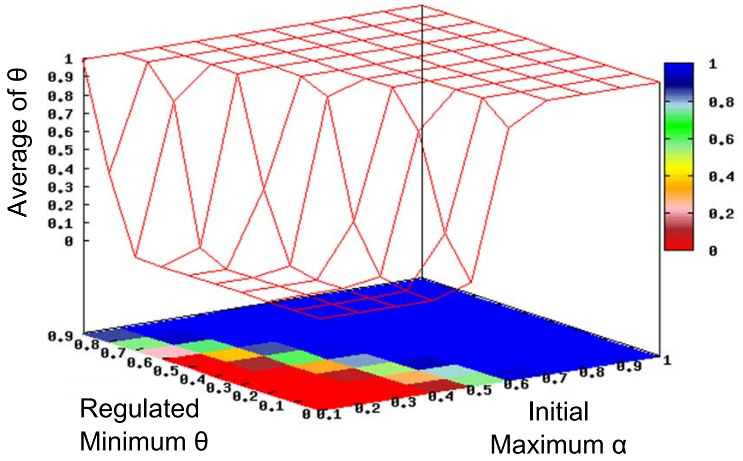


Fig. 5. The effect of regulation and the initial social state on the dissemination of CSR. The left axis is the minimum of the regulated invest ratio of the corporations, the right axis is the maximum of the initial importance ratio of the consumers, and the z axis is the average of the importance ratios of the consumers (averaged in 5 simulations).

5 Conclusion

In this paper, we study a mechanism to disseminate CSR activities, a normative action by corporations. The CSR is conceptualised as providing a quasi-public good simultaneously in addition to a private good. Using multi-agent simulations, we showed that the criteria to disseminate CSR were to take global information such as the average amount of investment in CSR into consideration to evaluate consumers themselves. The dissemination mechanism is a positive feedback between the consumers' social leaning of their preference putting importance on CSR and the corporates' popularity seeking. We also showed that by regulating the least amount of investment in CSR by corporates, dissemination could be launched even in an objectionable society.

References

1. Veblen, T.: The Place of Science in Modern Civilisation and Other Essays. Huebsch (1919)
2. Egashira, S., Hashimoto, T.: Status of human cognition in social science. In Nishibe, M., ed.: The frontier of evolutionary economics. Nihon-hyoron-sha (2004) 159–180
3. Besley, T., Ghatak, M.: Retailing public goods: The economics of corporate social responsibility. Journal of Public Economics **91** (2007) 1645–1663
4. Egashira, S.: Corprate social responsibility activities in japanese companies. Unpublished investigation (2009)

Partially Observable Markov Decision Processes with Behavioral Norms

(Extended abstract)

Matthias Nickles[1] and Achim Rettinger[2]

[1] Department of Computer Science, University of Bath
Bath, BA2 7AY, United Kingdom
m.l.nickles@cs.bath.ac.uk,
[2] Department of Computer Science, Technical University of Munich,
D-85748 Garching bei München, Germany
achim.rettinger@cs.tum.edu

Abstract. This extended abstract discusses various approaches to the constraining of Partially Observable Markov Decision Processes (POMDPs) using social norms and logical assertions in a dynamic logic framework. Whereas the exploitation of synergies among formal logic on the one hand and stochastic approaches and machine learning on the other is gaining significantly increasing interest since several years, most of the respective approaches fall into the category of relational learning in the widest sense, including inductive (stochastic) logic programming. In contrast, the use of formal knowledge (including knowledge about social norms) for the provision of hard constraints and prior knowledge for some stochastic learning or modeling task is much less frequently approached. Although we do not propose directly implementable technical solutions, it is hoped that this work is a useful contribution to a discussion about the usefulness and feasibility of approaches from norm research and formal logic in the context of stochastic behavioral models, and vice versa.

Keywords: Norms, Partially Observable Markov Decision Processes, Deontic Logic, Propositional Dynamic Logic

1 Introduction

This extended abstract discusses various approaches to the constraining of *Partially Observable Markov Decision Processes* (POMDPs) using social norms and logical assertions in a dynamic logic framework. Whereas the exploitation of synergies among formal logic on the one hand and stochastic approaches and machine learning on the other is gaining significantly increasing interest since several years, most of the respective approaches fall into the category of *relational learning* in the widest sense [12], including inductive (stochastic) logic programming. In contrast, the use of formal hard constraints and prior knowledge for other stochastic modeling or machine learning tasks is relatively seldom approached (”hard” constraint in the sense that the constraint cannot be overwritten, ignored or weakened). Among the existing approaches which allow for the specification of hard constraints of stochastic tasks are [5, 11], and, closer to our work, various extensions of Golog, such as [7, 6]. [7, 6] allow for the logic-based partial specification of a program and the automatic and optimal completion of this program, which is viewed as a Markov Decision Process (respectively a POMDP in case of [7]). In contrast to these approaches, we propose to take an ordinary POMDP (which could be manually created or automatically learned) and a set of "ordinary" modal logic formulas, and use the latter in order to modify the given POMDP and/or a standard algorithm for solving this POMDP (i.e., the search for an optimal behavior policy) so that certain action sequences become impossible or less probable (because they would violate a norm), or obligatory. Besides this, our approach uses a variant of dynamic logic, whereas Golog is based on the situation calculus. We find dynamic logic more useful for dealing with complex actions (patterns and compositions of elementary actions) than the situation calculus, and specifically for the specification of norms about complex actions.

We believe that the normative constraining of a stochastic model of an agent's environment would be useful for various potential applications. Agents in a multiagent system are potentially subject to social norms (respectively, sanctions in case of norm violating behavior) and need to take these norms into consideration when they plan their behavior. If the environment is noisy and/or only partially accessible to the agent's perception, or - more generally - if the agent is uncertain about the state of its environment, it needs to maintain a stochastic model of this environment and act in dependency from uncertain beliefs - including the compliance or intentional noncompliance with any of the norms. While it would be possible for the agent to consult its "norm base" (or to query the normative system in some way) at each step, it appears to be much more efficient to embed the knowledge about norms directly into the stochastic decision model of the agent, in terms of expected positive or negative rewards in case of norm obedience or failure to do so, respectively.

Although we do not propose directly implementable technical solutions in this paper, it is hoped that this work is a useful contribution to a discussion about the usefulness and feasibility of approaches from norm research and formal logic in the context of stochastic uncertainty modeling, and vice versa.

The remainder of this work is organized as follows: the next section describes the deontic logic we use to represent behavioral norms. Section 3 provides a brief introduction of POMDPs. Section 4 outlines and discusses various possibilities for the logical constraining of POMDPs using our formal framework. Section 5 concludes.

2 Representing Norms Using Dynamic Logic

We use *Propositional Deontic Logic* (PD_eL) [1] to represent the norms [10] which the agent is subject to as well as the agent's knowledge. PD_eL is a variant of

Propositional Dynamic Logic (PDL) [14, 2], and has as such properties which are very handy in our context: it is not only a well-researched language with a sound axiom system, but its Kripke-style semantics is also relatively close to the representation of Markov processes we will use later. More precisely, our semantics of PD_eL uses Kripke structures where the meaning of a certain construct is defined in terms of the current state and actions which define transitions to one ore more other states. We will use this later to constrain a Partially Observable Markov Decision Process (POMDP) [4, 8, 13]. Furthermore, PD_eL deals well with conflicting obligations and avoids several paradoxes known from other deontic logics [1].

And finally, it can be shown that certain description logics are syntactic variants of PDL [3], a fact which might be useful in order to represent PD_eL encodable norms and knowledge on the Semantic Web. However, we have not investigated this possibility in this work.

PD_eL is a normal modal logic K with additional axioms for actions. The only modal operator $[\alpha]$ has more or less the same meaning as in standard dynamic logic and corresponds to an action-annotated necessity multi-modality $\Box_{action}$. I.e., $[\alpha]\phi$ denotes a sufficient precondition for ϕ after action α has been performed. Obligation, permission and prohibition are derived from this modality together with a special proposition V which is used as a marker for undesirable states. Although this appears at a first glance as a kind of work-around compared to deontic logics with a dedicated deontic modality, it is actually an elegant solution which nicely reflects the agent's rationale for observing a norm and furthermore allows for a straightforward mapping of Kripke states to the reward-annotated (i.e., more or less desired) states of a POMDP. We are aware of the limitations arising from the fact that our logical framework (but of course not the POMDPs) only has neutral and undesirable states but cannot express positive rewards directly. That is, we can only model negative sanctions. Future versions might overcome this limitation.

We assume in this paper that there is a single agent which would be negatively sanctioned if it would not observe all given norms as far as possible (that is, under the provision that the respective desired states are reachable), and that all norms are equally preferred. However, there is no principled reason why these assumptions could not be dropped, at the price of a technically somewhat more complex model.

In the following, we present an abbreviated account of PD_eL; for full details please refer to [1]. First, we introduce the following sets:

- A non-empty set *Act* of action expressions. Although PD_eL is not identical with PDL, we make use of the PDL terminology and refer to the elements of *Act* as programs.
- A non-empty set Act_0 of atomic actions.
- A non-empty set Φ of formulas.

which are the smallest sets satisfying the following conditions (with $\alpha, \alpha_1, \alpha_2 \in$ *Act*, $\phi, \phi_1, \phi_2 \in \mathit{\Phi}$):

1. $A_0 \subset \mathit{Act}$
2. $\emptyset \in \mathit{Act}$, $\mathit{Any} \in \mathit{Act}$ ($\emptyset$ stands for "failure" (an impossible action with no successor state), *Any* for "any actions" (some non-deterministically choosen atomic actions are performed simultaneously.))
3. $\alpha_1; \alpha_2 \in \mathit{Act}$ (sequential composition)
4. $\alpha_1 \cup \alpha_2 \in \mathit{Act}$ (choice. Perform either α_1 or α_2.)
5. $\alpha_1 \& \alpha_2 \in \mathit{Act}$ (joint action. Perform α_1 and α_2 concurrently.)
6. $\phi \rightarrow \alpha_1/\alpha_2 \in \mathit{Act}$ (conditional action. If ϕ holds in the current state, perform α_1, and α_2 otherwise.)
7. $\overline{\alpha} \in \mathit{Act}$ (negated action. Not action α)
8. $V \in \mathit{\Phi}$ (the special proposition which marks "unpleasant states")
9. $\phi_1 \vee \phi_2, \phi_1 \wedge \phi_2, \phi_1 \rightarrow \phi_2, \phi_1 \equiv \phi_2, \neg\phi \in \mathit{\Phi}$
10. $[\alpha]\phi, <\alpha> \phi \in \mathit{\Phi}$ (with $<>$ being the dual of $[]$, with $\sigma \models <\alpha> \phi \Leftrightarrow_{def} \sigma \models \neg[\alpha]\neg\phi$)

Norms can be specified using the following abbreviations:

Prohibition $\sigma \models F\alpha \Leftrightarrow_{def} \sigma \models [\alpha]V$ (we say that it is *forbidden* to do α)
Obligation $\sigma \models O\alpha \Leftrightarrow_{def} \sigma \models F\overline{\alpha}$ (we say that the agent is *obliged* to do α)
Permission $\sigma \models P\alpha \Leftrightarrow_{def} \sigma \models \neg F\alpha$ (we say that it is *permitted* to do α)

2.1 Semantics and Axioms of PD_eL

The Kripke-style semantics, based on the semantics of PDL, is outlined in Section 4.1.

Axioms (in addition to those of propositional logic):

$$[\alpha](\phi_1 \rightarrow \phi_2) \rightarrow ([\alpha]\phi_1 \rightarrow [\alpha]\phi_2) \tag{1}$$

$$[\alpha_1; \alpha_2]\phi \equiv [\alpha_1]([\alpha_2]\phi) \tag{2}$$

$$[\alpha_1 \cup \alpha_2]\phi \equiv [\alpha_1]\phi \wedge [\alpha_2]\alpha \tag{3}$$

$$[\alpha_1]\phi \vee [\alpha_2]\phi \rightarrow [\alpha_1 \& \alpha_2]\phi \tag{4}$$

$$[\phi_1 \rightarrow \alpha_1/\alpha_2]\phi_2 \equiv (\phi_1 \rightarrow [\alpha_1]\phi_2) \wedge (\neg\phi_1 \rightarrow [\alpha_2]\phi_2) \tag{5}$$

$$< \alpha > \phi \equiv \neg[\alpha]\neg\phi \tag{6}$$

$$[\overline{\alpha_1; \alpha_2}]\phi \equiv [\overline{\alpha_1}]\phi \wedge [\alpha_1][\overline{\alpha_2}]\phi \tag{7}$$

$$[\overline{\alpha_1}]\phi \vee [\overline{\alpha_2}]\phi \rightarrow [\overline{\alpha_1 \cup \alpha_2}]\phi \tag{8}$$

$$[\overline{\alpha_1 \& \alpha_2}]\phi \equiv [\overline{\alpha_1}]\phi \wedge [\overline{\alpha_2}]\phi \tag{9}$$

$$[\overline{\phi_1 \rightarrow \alpha_1/\alpha_2}]\phi_2 \equiv (\phi_1 \rightarrow [\overline{\alpha_1}]\phi_2) \wedge (\neg\phi_1 \rightarrow [\overline{\alpha_2}]\phi_2) \tag{10}$$

$$[\overline{\overline{\alpha}}]\phi \equiv [\alpha]\phi \tag{11}$$

$$[\emptyset]\phi \tag{12}$$

3 Partially Observable Markov Decision Processes

Partially Observable Markov Decision Processes (POMDPs) model an agent's decision problems in some environment where the agent's perception is limited or noisy [4, 8, 13]. The primary goal of the agent is to find an optimal action policy, that is, to find a sequence of decisions regarding its behavior such that its reward is maximized.

Formally, a POMDP is a tuple $(S, A_0, T, R, O, \varOmega)$, where

- S is a finite, non-empty set of world states, denoted in this paper as "states" or "Markovian states" (the latter in demarcation from the larger set of world states used in the Kripke structures),
- A_0 is the finite set of atomic actions,
- $T : S \times A_0 \to \varPi(S)$ is the state-transition function. For each state and each (atomic) agent action $a \in A_0$ it yields a probability distribution over states. $T(\sigma, a, \sigma')$ stands for the probability that the agent ends in state σ' given it starts in state σ and performs atomic action a.
- $\varOmega$ is the set of all possible observations the agent can make in its environment.
- $R : S \times A_0 \to \mathbb{R}$ is the agent's reward function. It yields for each action and each state the immediate reward $R(s, \alpha)$ for taking this action.
- $O : S \times A_0 \to \varPi(\varOmega)$ is the observation function, which gives for each action and each resulting state a probability distribution over possible observations. $O(\sigma', a, o)$ stands for the probability of making observation o after performing atomic action $a \in A_0$ and ending with this action in state σ'.

We use the same symbol A_0 for the set of atomic actions in PD_eL, since both sets are actually identical in our framework. S should correspond to a set of states (worlds) in the Kripke-structures (cf. the next section).

The next state and the reward depend only on the current state and the performed action. That is, POMDPs fulfil the Markov property.

A POMDP models an uncertain part of the agent's subjective and dynamic beliefs about a noisy environment in which the agent takes action. However, we do not make this explicit in our *logical* framework (which would require us to introduce inter alia a doxastic modality and a probability distribution over states, as in, e.g., [9]). Instead we will later update a given, inaccurate POMDP with certain knowledge from our PD_eL knowledge base (KB). The KB and the POMDP can then either co-exist, or only the POMDP is maintained.

Given a POMDP, the agent's tasks are i) to update its belief state in dependency from its previous belief state, the agent's current observation, and the agent's last action and ii) to generate optimal actions, depending on the belief state and expected rewards.

A belief state is a probability distribution over world states. It can be seen as a roundup of the agent's initial belief state updated by its past experiences. Because of this, it is not required to take into account the history of past actions and observations explicitly for decision making. However, there are infinitely many belief states.

Formally, a belief state is a function $b : S \times [0;1]$ and $b(s)$ is the probability that the agent is in state s, with $\sum_{s \in S} b(s) = 1$.

Computing a new belief state $b' = \tau(b, \alpha, o)$ given the old belief state b, an action α and an observation o (*state estimation*) is not very complicated. $\tau(b, \alpha, o)$ is called the *belief state transition function.*

$$b'(s') = Pr(s'|o, \alpha, b) \tag{13}$$

$$= \frac{Pr(o|s', \alpha, b) Pr(s'|a, b)}{Pr(o|\alpha, b)} \tag{14}$$

$$= \frac{Pr(o|s', \alpha) \sum_{s \in S} Pr(s'|\alpha, b, s) Pr(s|a, b)}{Pr(o|\alpha, b} \tag{15}$$

$$= \frac{O(s', \alpha, o) \sum_{s \in S} T(s, \alpha, s') b(s)}{Pr(o|\alpha, b)} \tag{16}$$

The belief states together with their updating function form a certain kind of *observable* Markov Decision Process, a so-called *continuous state space belief-MDP*. This insight is crucial for solving a POMDP, since it allows to formulate the solution of the POMDP (i.e., the optimal behavioral policy) as the solution of this kind of MDP.
The belief-MDP is defined as a tuple (B, A_0, τ, r), with:

- B being the set of belief states, as defined above,
- A_0 being the same action set as for the POMDP,
- τ being the belief state transition function, and
- $r : B \times A_0 \to \mathbb{R}$, the reward function of the belief states, with $r(b, a) = \sum_{s \in S} b(s) R(s, a)$ (R is the agent's reward function as defined for the POMDP, i.e., operating on actual world states instead of uncertain beliefs about such states).

The so-called optimal *value function* V^* finally yields the agent's subjective value of being in a certain belief state. Many POMDP solving approaches compute or approximate this function using dynamic programming updates of sub-optimal value functions and derive from the optimal (or good enough) value function the optimal (or good enough) action policy (e.g., [4, 13]). There are also algorithms which search the space of policies directly for the optimal policy (e.g., [8]). The latter type of algorithms nevertheless also requires to know the corresponding value functions of policies, in order to evaluate policies and to single out the optimal policy (or a good enough approximation). The following recursive definition of V^* is called the dynamic programming equation of the POMDP (γ is a discount factor):

$$V^*(b) = max_{\alpha \in A_0}(r(b, a) + \gamma \Sigma_{o \in O} Pr(o|b, a) V^*(\tau(b, a, o))) \tag{17}$$

Unfortunately, the belief-MDP is over a continuous state space, which poses various problems. But fortunately, POMDP solvers can exploit the fact that the

MDP for which the optimal value function is a solution is a converted POMDP, a fact which yields certain useful properties of the function.

4 Modal-Logical Constraining of POMDPs

We assume a given POMDP and a knowledge base (KB) of PD_eL assertions. The task of combining these two in order to retrieve a new, normatively and assertively constrained POMDP is twofold:

- Obtaining a constrained belief estimator from prior knowledge in the KB and
- pruning the set of potentially optimal action policies when solving the POMDP in order to observe the the norms encoded in the KB.

4.1 Assigning Propositions to Markovian States

The states of a POMDP don't tell us anything about the values of the propositional variables in the respective states. In contrast, the KB basically tells us which propositions hold after a certain program has terminated. A Kripke model $\mathfrak{K}$ is defined by $\mathfrak{K} = (K, m_{\mathfrak{K}}, \models)$, with $(K, m_{\mathfrak{K}})$ being the Kripke frame consisting of the set of world states K, and the meaning of each atomic formula and each atomic action, given as a mapping $m_{\mathfrak{K}}$ of this formula/action to a subset of the world states (that is, the states where the formula holds) or set of pairs of world states, respectively (that is, the "input state" which is mapped to the "output state" via an action). $m_{\mathfrak{K}}$ can be extended inductively to work with any formula and complex actions (programs) too.

Formally:

$$m_{\mathfrak{K}}(\psi) \subset K \textit{ foreach } \psi \in \Phi, \textit{ and} \tag{18}$$

$$m_{\mathfrak{K}}(\alpha) \subset K \times K \textit{ foreach } \alpha \in Act \tag{19}$$

With this, we can define the semantics of PD_eL formulas based on the semantics of PDL [2], like:

$$\sigma \models [\alpha]\psi \Leftrightarrow_{def} \forall \sigma' : \textit{if } (\sigma, \sigma') \in m_{\mathfrak{K}}(\alpha) \textit{ then } \sigma' \models \psi \tag{20}$$

Theoretically, we could use this semantics directly i) to update the POMDP state transition matrix with definite transitions, ii) to set an element (subjective state probability) of a POMDP belief state to zero if the respective state would be logically impossible, and iii) to gain knowledge about the values of proposition variables after each belief update. In case i) and ii), the given POMDP is treated as possibly partially invalid, and the invalid parts are precisely those which are "overwritten" with definite prior knowledge deductively obtained from the KB. We treat approaches iii) and i) as mutually exclusive: iii) "believes" the result of the POMDP state estimation, whereas i) possibly extinguishes a result of the probabilistic state estimation.

ii) is expressed as follows:

$$\textit{if } \sigma \models [a]\psi, a \in A_0 \textit{ and } \sigma' \not\models \psi \textit{ then } b'(\sigma') = 0, \tag{21}$$

with $b' = \tau(b, a, o)$, for any observation o. In addition, a re-normalization of the probabilities of the other states is required, so that the sum of the probabilities becomes 1 again - which means that it is not possible to make all states impossible states at the same time!

For iii), we need to extend our notion of belief states to *logically-annotated belief states* $b_\Phi : S \times ([0; 1] \times \Phi)$. Then we have the rule

$$\textit{if } \sigma \models [a]\psi, a \in A_0 \textit{ then } b'_\Phi(\sigma') = (\frac{O(\sigma', a, o) \sum_{\sigma \in S} T(\sigma, a, \sigma') b(\sigma)}{Pr(o|a, b)}, \psi), \tag{22}$$

for any observation o.

The retrieval of formal knowledge about a possible state during state estimation can be useful for the acting agent, provided it can interpret the logical annotations.

i) can be expressed using the rule

$$\textit{if } \sigma \models [a]\psi, a \in A_0 \textit{ and } \sigma' \not\models \psi \textit{ then } T(\sigma, \alpha, \sigma') = 0 \tag{23}$$

(again, this would require normalization of the belief state to make it represent a probability distribution).

Practically, it would make sense to consider only formulas which hold in *all* states:

$$\models \psi \Leftrightarrow_{def} \forall \sigma \in K : \sigma \models \psi$$

The constraining of belief updates does then not depend on the respective previous states anymore.

But still the approaches i)-iii) have obviously two shortcomings: firstly, we do not know the mapping of Markovian states to states in K. Secondly, they work only with atomic actions.

The first problem could be solved by considering POMDPs with logically-annotated Markovian states (not to be confused with the logically-annotated belief states above). If we would annotate a subset of the states with a set of formal assertions each, we could nullify those parts of the current belief state which are logically inconsistent w.r.t. the KB. Let $\phi : S \rightarrow 2^\Phi$ be a function which maps a Markovian state to a (possibly empty) set of assertions which are *known* by the agent to hold in this state. Then

$$\textit{if} \;\; \models [a]\psi, a \in A_0 \; \textit{and} \;\; \models \neg(\psi \wedge \bigwedge_{f \in \phi(\sigma')}) \; \textit{then} \; b'(\sigma') = 0, \tag{24}$$

with $b' = \tau(b, a, o)$, for any observation o and any previous belief state b.

Getting rid of the second shortcoming would be a bit more tricky: we deal with state transitions instead of action histories: each Markovian state is a sufficient statistics in the sense that the probability distribution of successor states depends only on this state (and the current action and observation) and not on any preceding states or action history.

4.2 Constraining the Optimal Action Policy using Norms

Each atomic actions sequence which leads to a world state where the special proposition V holds should be removed from the set of candidates for the optimal action policy, or the value (utility) of such states should be reduced. With this, it becomes more unlikely than otherwise (but not necessarily impossible) that the solution of the POMDP tells the agent to run into a norm-violating state.

In the most simple case, the agent is forbidden to take a single atomic action. In POMDP terms, this can be taken into account by reducing the respective reward of performing this action in any state:

$\textit{if} \;\; \models Fa, a \in A_0 \; \textit{then} \; \forall\sigma \in S : R(\sigma, a) = \varrho$. Here, the reward is simply set to some negative value ϱ in order to make action a less desirable.

We could alternatively annotate all states but one which are reachable via a with $\neg V$. However, a POMDP does not allow us to make a certain action always lead to an "impossible" Markovian state.

The general case of prohibited (obligatory, allowed) complex actions is significantly more complicated:

The sequences of atomic actions described by a certain action expression can be represented as so-called *s-traces* (*synchronicity traces*) [1]. To represent the set of all s-traces for a certain action expression, we use the notation $[[\alpha]]$. The exact definition of this function can be found in [1]. Each s-trace $s \in [[\alpha]]$ is a sequence $S_1, ..., S_n, ...$ of so-called *synchronicity sets* (*s-sets*) S_i. A single s-set if a subset of Act_0. Intuitively, a single s-set represents a number of atomic actions which are performed concurrently (if the set contains more than one action), or a single atomic action. We call each possible sequence of atomic actions within $[[\alpha]]$ a *run* of α.

To enact the prohibition of a complex action α using its s-traces, we propose the following alternative approaches:

1. Modify the optimal policy directly, in order to make the execution of any action sequence within $[[\alpha]]$ impossible.
2. Modify (decrease) the values of the belief states which are reachable using action sequences in $[[\alpha]]$.
3. Modify (lower) those vectors which contribute to the value function computed during value or policy iteration and which represent an action within an action sequence in $[[\alpha]]$ (see below).

Both policy search and value function search algorithms for solving a POMDP require the computation of value functions (cf. Section 3), from which the optimal policy can be derived directly. For approach 1, we assume that the optimal policy (ignoring norms) is already given, in form of a finite-state machine (FSM). A FSM can always be used to represent the optimal behavioral policy of a *finite-horizon* POMDP, which appears to be a reasonable restriction in our context [8].

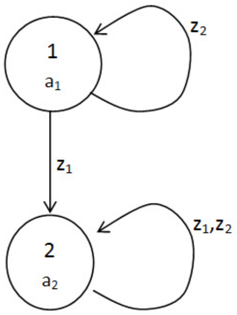


Fig. 1. A simple FSM representing a policy

Figure 1 depicts a FSM which represents a policy for some POMDP as follows: each node (FSM state, labeled with a number inside of the respective node) is annotated with an action a_i which the agents takes in this FSM state. If the policy is optimal, this action is optimal in the respective FSM state. State 1 is the start state. Each arc represents a possible observation z_i following the respective action and leads to a new FSM state. There can be more FSM states than world states.

Assume we have $\models F(a_1; a_1); a_2$. The s-trace of the forbidden program would then simply consist of a single, deterministic run of atomic actions. Removing this sequence from the FSM could yield the new FSM depicted in Figure 2 (with the dashed arc not being part of the FSM). Of course, this FSM is just one

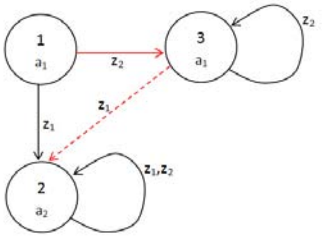


Fig. 2. An updated FSM representing a policy with forbidden paths

among many possible updated FSMs. The major shortcoming of approach 1 is obviously that the agent is not prevented from starting a forbidden action sequence. The sequence is simply cut off before it finishes and makes the agent stop then. Re-directing the agent to some random state instead appears not to be an improvement.

Approach 2 requires us to keep track of actions during the iterative belief state updates, provided the used POMDP solving algorithm allows us to do this. We assume again that $\models F(a_1; a_1); a_2$. The set of belief states the agent might end in after performing run $a_1 \circ a_1$ is computed as $B_{a_1 \circ a_1} = \{\tau(\tau(b_s, a_1, o_1), a_1, o_2)$ for all possible observations o_i and initial belief states $b_s\}$.

After(!) having computed the optimal value function, the agent can incrementally update its belief state and compute at each step the optimal action from the optimal value function. Should the agent run into one of the belief states in $B_{a_1 \circ a_1}$, and the optimal action policy (ignoring norms) suggests to take action a_2 next, it could, as with approach 1, avoid doing so. Again, this approach is rather unconvincing. To be more interesting appears a reduction of the values of the belief states $\{\tau(\tau(\tau(b_s, a_1, o_1), a_1, o_2), a_2, o_3)\}$ during the search for the optimal value function. This way, the agent is more likely to be prevented of running into a forbidden sequence of acting, since the value of a certain state includes the values of succeeding states.

Finally, approach 3 makes use of the fact that each state of a (possibly yet sub-optimal) FSM representing a policy during the search for an optimal policy (using one of the POMDP solvers which search the policy space directly for the optimal policy) corresponds to a vector $v^i(s)$ of the piecewise linear and convex value function which can be computed from this FSM under certain conditions [8]. The value function is the solution of the following system of equations, with one equation for each pair of FSM state i and Markovian state σ:

$$v^i(\sigma) = R(\sigma, a(i)) + \gamma \sum_{\sigma', z} Pr(\sigma'|\sigma, a(i)) Pr(z|\sigma', a(i)) v^{l(i,z)(\sigma')} \tag{25}$$

Performing a forbidden sequence of atomic actions as determined by the FSM and observations yields a resulting FSM state which corresponds to exactly one of these vectors v^i, which is associated with the optimal action in this state. Lowering this vector, i.e., modifying the value function at this place, would lower the value of this FSM, and would, as we assume, lead in the next policy search step to a FSM which is closer to norm-observing behavior. The advantage of this approach is that from a modified set of vectors (which is then in addition also improved by a dynamic programming update) a new FSM can be constructed very easily [8]. However, this approach would require extensive experimental evaluation in order to judge whether it would actually make sense in a concrete scenario.

5 Conclusion

In this extended abstract we have proposed various initial means for the embedding of knowledge about norms and formal knowledge in general into POMDPs, hoping to initiate a new line of future research. Although we have hopefully given some initial insight into the challenge, a lot of work remains to be done:

- Identification of application scenarios which are on the one hand rich enough to allow for a more or less realistic normative system, but which are on the other hand still approachable by contemporary POMDP solver.
- Detailed empirical and theoretical analysis of the constraining task, with a detailed comparison of the proposed and further ways of incorporating norms into POMDPs and stochastic decision processes in general.
- Detailed empirical and theoretical analysis of how the constraining affects the POMDP solving algorithm.
- Consideration of more complex kinds of norms, such as norm hierarchies and preferences among norms.

References

1. J.-J. Meyer. A Different Approach to Deontic Logic: Deontic Logic Viewed as a Variant of Dynamic Logic. Notre Dame Journal of Formal Logic, 29, 1988.
2. D. Harel, D. Kozen and J. Tiuryn. Dynamic Logic. MIT Press, 2000.
3. P. Blackburn, J. van Benthem, F. Wolter (Eds.). Handbook of Modal Logic. Elsevier, 2006.
4. L. P. Kaelbling, M. L. Littman, A. R. Cassandra. Planning and Acting in Partially Observable Stochastic Domains. Artificial Intelligence 101(1-2): 99-134, 1998
5. K. Wagstaff, C. Cardie. Clustering with Instance-level Constraints. In Proceedings of the 17th International Conference on Machine Learning (ICML 2000), 2000.
6. C. Boutilier, R. Reiter, M. Soutchanski, S. Thrun. Decision-Theoretic, High-level Agent Programming in the Situation Calculus. AAAI 2000, 2000.
7. A. Farinelli, A. Finzi, Th. Lukasiewicz. Team Programming in Golog under Partial Observability. In Proceedings of the 20th International Joint Conference on Artificial Intelligence (IJCAI 2007). AAAI Press / IJCAI, 2007.

8. E. Hansen. Solving POMDPs by Searching in Policy Space. In Proceedings of the Fourteenth International Conference on Uncertainty In Artificial Intelligence (UAI-98), 1998.
9. F. Fischer, M. Nickles. Computational Opinions. In Proceedings of the 17th European Conference on Artificial Intelligence (ECAI-06), IOS Press, 2006.
10. G. Boella, M. Singh, G. Pigozzi, H. Verhagen (Eds.). Proceedings of the Third International Workshop on Normative Multiagent Systems (NorMAS), 2008.
11. I. Davidson, S. S. Ravi. The complexity of non-hierarchical clustering with instance and cluster level constraints. Data Mining and Knowledge Discovery 14(1), 2007.
12. L. Getoor, B. Taskar (Eds.). Introduction to Statistical Relational Learning. The MIT Press, 2007.
13. M. Littman. Solving Partially Observable Markov Decision Processes via VFA. In J. A. Boyan, A. W. Moore, R. S. Sutton (Eds.), Proceedings of the Workshop on Value Function Approximation, Machine Learning Conference 1995, Technical report CMU-CS-95-206, 1995.
14. M. Fischer, R. Ladner. Propositional Dynamic Logic of Regular Programs, Journal of Computer and System Sciences, 18: 194-211.

Reflection and Norms: Towards a Method for Dynamic Adaptation for MAS

Ingo J. Timm, Andreas D. Lattner, René Schumann

Information Systems and Simulation, Goethe University Frankfurt
Robert-Mayer-Str. 10, 60325 Frankfurt am Main, Germany
timm|lattner|reschu@cs.uni-frankfurt.de

Abstract. The design of self-organizing systems and particular multiagent systems (MAS) is a non trivial task. On the one hand the particular system should show a dynamic behavior according to its environment, to gain a central advantage of distributed systems, on the other hand it has to act on behalf of its user and the final results have to possess acceptable quality. Especially the quality of the overall system's behavior can become a critical issue, if the subsystems have their own objectives they have to optimize. In this paper we present a methodology that can be integrated into MAS for adapting their behavior allowing local optimization while respecting an acceptable level of the system's global goals.

Keywords. balancing autonomy, multiagent simulation, manufacturing control

1 Introduction

In the last years, a trend towards decentralized designs of decision support and decision taking systems can be observed. In contrast to the centralized systems, it is in practical cases not possible to compute optimal results according to a global objective function. Resulting plans and schedules will be suboptimal [1,2]. Even so decentralized control systems are in the focus of current discussion and research. The reason is that (cf. [3])

- systems with higher flexibility and reliability can be designed and
- decentralized control becomes part of current company organization.

But it turns out to be hard to design systems that on the one hand show a flexible behavior and on the other hand act on behalf of the user and reach acceptable solution quality in comparison with e.g. centralized solution approaches. This is especially the case if the entire systems comprise of subsystems that have a local objective they try to optimize. Note that this is a typical situation if the entire system was developed following the divide and conquer engineering paradigm.

The idea of self-organizing systems have attracted attention by researchers that try to overcome this design problem by allowing the system to self-configure

Dagstuhl Seminar Proceedings 09121
Normative Multi-Agent Systems
http://drops.dagstuhl.de/opus/volltexte/2009/1917

itself to the given environments and given objective functions. In the following we look only at such systems. That is each agent has its local goals and objective function that it tries to optimize while achieving its goals. Moreover the agents form a MAS that have to fulfill global goals and to optimize a global objective function within a dynamic environment. As a consequence of the complex problem and the dynamic environment a flexible solution, e.g. based on a MAS, has advantages to classical centralized optimizing approaches. Norms have been identified to be a valuable method to allow multiagent systems (MAS) flexible behavior still having a hand on the overall system behavior. But the norm design and system configuration respectively calibration can become a complex and time consuming issue. The idea of dynamic adaptation is that the system adjusts itself seamlessly to the given situation [4]. In this paper we outline a method that can be integrated into a MAS that gives the ability of dynamic adaptation respecting local and global goals and objectives. Local entities can perform their actions, and if necessary adapt, enforced or not, their behavior towards the global objective. This step is called strategy revision. If this revision is not sufficient for some reason, a more general scheme is applied. Based on sociologically concepts we propose a reflection phase that allows to change the existing strategy revision strategies or even objectives. Within this method norms can play an important role, as they can not only be used to control the system directly, but can be used guiding the reflection phase as well.

As a step within this methodology we present one possible implementation of the strategy revision step, called regulated autonomy [5]. This is a centralized rule-based implementation of a strategy revision step. For the evaluation of this approach we use a very simple manufacturing scenario. A simple scenario is chosen to provide competitive results of this technology with other classical approaches, that typically cannot be applied to complex problems as they can be found in practice.

This paper is structured as follows. In the next section we outline related work concerning with the notion of autonomy in MAS and the adaption of autonomy at runtime. The related work for norm-based systems is left open for further discussion. In section 3 the principles of reflections are presented. The concept of regulated autonomy, already developed as an efficient implementation of a strategy revision strategy is presented in section 4. Finally we summarize and sketch possible issues to be addressed in future research.

2 Related Work

In this section we review existing work on autonomy and adaption of autonomy in MAS. The special focus of norms is left open for further discussions during the workshop.

2.1 Autonomy in MAS

In the literature, there are discussions on different levels of autonomy [6,7]. The definition of autonomy in the literature ranges from very wide autonomy [8] to restricted cases [9].

In early multiagent research, Castelfranchi and Conte [8] discuss a very high degree of autonomy, such as the influence of predefined norms, behavior patterns, or procedures is irrelevant, and the relevance is very low with respect to the action-selection process within an agent, respectively.

A more theoretical and interdisciplinary approach to define the term autonomy was presented by Bertschinger et al. [10]. Their key question is how autonomy can be measured. Therefore, they use an information theoretic perspective which bases on the distinction between the system and its environment. Depending on the ability of the system to influence the environment, Bertschinger et al. present different metrics for autonomy.

Additionally, the scope of autonomy – or more precisely the scope or the context autonomy refers to – is discussed by Kirn [11]. Kirn points out that autonomy can address different aspects:

- association autonomy (decision freedom towards the participation of agent societies)
- cooperation autonomy (towards the participation in cooperative processes)
- execution autonomy (local execution)
- resource autonomy (disposition of local resources)
- communication autonomy (this is the participation in communication)

In each of these aspects, the autonomy can be specified for the agent.

Bradshaw et al. [12] define the term autonomy in respect to their research about adjustable autonomy. From their perspective, autonomy is mainly characterized by two aspects:

- self-sufficiency, as the capability of an entity to take care of itself and
- self-directedness, as the freedom from outside control.

According to Bradshaw et al., autonomy can be related to two dimensions. A descriptive dimension which describes if the agent is capable to perform an action and a prescriptive dimension describing if the agent is allowed to perform an action.

Barber and Martin [13] define and measure autonomy as the interdependency of an agent in its decision making to achieve its goals. An agent is autonomous if it is capable to pursue some goals without interference by other agents.

Luck et al. [14] present a strong definition of autonomy. According to them the self-generation of goals is the defining characteristic about autonomy. These goals are generated or derived from motivations an agent has encoded.

Schillo [15,16] introduces a "Framework for self-Organization and Robustness in Multiagent systems" (FORM) where delegation is the main concept in order to describe organizational relationships. He distinguishes between task and social delegation and four different mechanisms for these two delegation modes (Economic Exchange, Gift Exchange, Authority, and Voting). He defines a spectrum of seven organizational forms for groups of agents by using the delegation types and modes as building blocks: Single Autonomous Agents, Task Delegation, Virtual Enterprise, Cooperation, Strategic Network, Group, and Corporation.

Nickles et al. [17] present a specification schema for computational autonomy based on sociological role theory, namely RNS ("Roles, Norms, Sanctions"), in order to "support developers of agent-oriented applications in specifying the kind and level of autonomy (...)". The viewpoint of the authors is that agents act as role owners encountering certain norms in a social frame which regulates the behavior of the agents. In RNS, three types of norms ("permissions, obligations, and interdictions") as well as two sanction types ("reward and punishment") are distinguished [17]. The sanctions can be specified explicitly by the designer and thus provide means to control the autonomy of the agents.
Nevertheless, autonomy is a property, which may lead to partially unwanted system states resulting from conflicting or inconsistent goal sets. The dynamic and complex interdependencies of autonomous subsystems can lead to systems whose organization emerges at runtime. Thus, software engineers of autonomous systems may not consider any possible constellation of subsystems at design time.

2.2 Runtime Adaptation of Autonomy in MAS

The runtime adaptation of autonomy in MAS is addressed by researchers with different application scenarios in mind, thus different terminologies evolved.
From the research about mixed-initiative interactions where agents and humans work together, mostly agents are guided by human operators. In this area the term *adjustable autonomy* has been established. Work about adjustable autonomy can be found, for instance, in [18,15,19,12]. Here we detail the approach by Bradshaw et al. [12]. The goal of adjustable autonomy is to maximize the opportunities for local adaption to unforseen problems and opportunities while assuring humans that agent behavior will be kept in desired bounds. Therefore different adaptations are possible:

- adjusting permission, add or remove rights
- adjusting obligations, add or remove tasks
- adjusting possibilities, add or remove skills
- adjusting capabilities, add or remove resources

These adjustments are done by the human operator, at his will.
With a slight different notion, but with the human-agent interaction focus Urbig and Schröter [20] describe the concept of "dynamic degrees of delegation" from "Full Autonomy to Manual Control". The work is based on the C-IPS approach which addresses different aspects of negotiation decisions in agents (C-IPS stands for external constraints (C), negotiation issues (I), partners (P), and negotiation steps (S)). Basically, agents would act with full autonomy. In order to let the user control the agent, Urbig and Schröter introduce means to distinguish between situations where the user should be involved and situations that can be handled autonomously by the agent. The degree of delegation can be specified for different decision types and dynamically changed during run time.
In the context of robotics, different levels of autonomy are discussed by [21].

Gancet and Lacroix define five levels of autonomy and define for each level which abilities and permissions a robot has with the given autonomy level.
Mailer [22] addresses the area of distributed problem solving where he found that it is useful to dynamically centralize the solving of overlapping subproblems in order to find solutions more quickly. He calls this approach mediation-based as it combines techniques from centralized and decentralized problem solving.
Barber et al. [23] present the concept of "Dynamic Adaptive Autonomy". This allows agents to switch autonomy within a defined spectrum. Thereby according to their notion of autonomy, presented above. The degree of autonomy depends on the goal. Thus for different goals of an agent, it can have different levels of autonomy.

Our approach is also of the second kind allowing a runtime adaptation of autonomy during runtime. An superior entity can force agents to follow certain strategies or to perform actions needed to obtain the desired overall system's performance.

3 Reflection

As mentioned in the introduction, there are challenges in engineering multiagent systems with respect to their properties of autonomy and interaction. The key reasons for applying agent technology in complex economical environments can be found in the high level of modularization and information hiding as well as potential for positive emergent effects. The question arises, if such an emergent behavior, i.e., a macroscopic behavior on the basis of microscopic interactions, is beneficial for the global system. In the beginning of multiagent research, this assumption was stated as a fact. Recent research focuses on sophisticated design of the autonomous subsystems to enable a positive effect of the whole system [24,25]. De Wolf and Holvoet [26] propose an approach for engineering self-organizing systems. Their approach is based on the analysis of the system after implementation and before delivery. Because of the well-known complexity of testing concurrent systems, the approach seems to be adequate for systems with a moderate amount of internal states, where no extensive internal states or static strategic behavior exists. However, sophisticated engineering is required to ensure the desired behavior on the basis of autonomous systems respectively balancing microscopic and macroscopic behavior.

In social science, the phenomena of microscopic-macroscopic interaction is widely researched [27,28,29]. Norms and regulations are introduced in a social system to establish a better system performance. In the following we introduce a methodology which is based on results from social science by Luhmann [30]. The theory focusses on reorganization processes in societies and specifies different steps of individual and social reflection. In this work, a system has the ability to reflect about its overall performance explicitly within negotiations. In an interdisciplinary research [1], we conceptualized a social mechanism as an explanatory model for societies based on the work of [30]. On this basis, a conceptual model for reflection in multiagent systems has been developed.

The multiagent conceptualization of reflection is based on the assumption that a multiagent system was chosen deliberately as a system design. In consequence, autonomy of the agents is not a side effect but one of the key features. If dependability on the multiagent level is in question, then some dynamic mechanism is required, which allows for context-dependent adjustment of the individual behavior. As the autonomy is a key feature, the adjustment of the individual behavior should be as restricted as possible. Furthermore, we aim at a reflection methodology, which preserves local autonomy even by global adjustment.

In economical systems we know these mechanisms for a long time. In the last decades, there is a trend in business administration manage people by delegating tasks with the definition of the context rather than supervising each step of execution. If the context or the boundaries are not met, the management is involved again to handle the exception. Keeping this in mind, the methodology of reflection consists of four steps as illustrated in Figure 1.

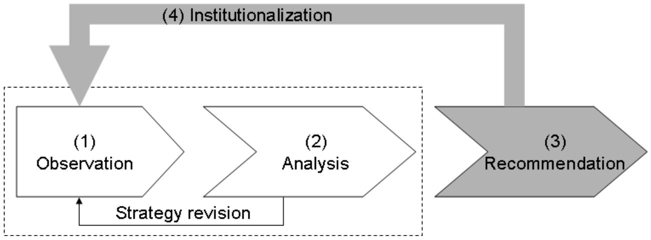


Fig. 1. Reflection in multiagent systems

The process of reflection is divided in two different parts. In the observation and analysis step, the individual agents are reflecting their behavior with respect to the global goals. The step of joint solution and institutionalization includes a group of agents which try to solve a problem cooperatively.

The approach is based on a group of agents which can be formed dynamically in runtime or specified at design-time. In either way, it is assumed, that the group of agents have some common goals and the fulfilment of this goal can be measured by the group or some other instance with communication capabilities. Furthermore, for each goal, there are different levels of goal satisfaction, i.e., 0 implies that a goal is completely unsatisfied and 1 indicates that the goal has been satisfied. For utility-based goals a continuous scale is assumed while for logic-based goals the goal-satisfaction is a binary function. Additionally, we assume that the consideration of global goals in every deliberation step would be inadequate with respect to computational or memory consumption. The basic assumption is that global system goals are known and that it is possible to evaluate a situation to what extent the target setting is achieved, i.e., a system has the ability to reflect about its overall performance explicitly within negotiations. It is distinguished between three color codes: target accomplished ("green"), target slightly failed ("yellow"), and target failed ("red").

The global context emerges from a cooperation or a collaboration. In a cooperation, agents does not have to fulfill the criteria of individual rationality as introduced by Sandholm [31], i.e., agents may ”suffer” by a joint solution without compensation. In collaborations, i.e., in settings, where competing agents are cooperating for a specific task, process, or time period, it is important, that the solution strategy is modeled in such a way, that each agent meets individual rationality, i.e., the agents are better off participating in the group than not participating.

In the observation stage, each agent reports its performance to a blackboard or central entity (group coordinator) within the group. The blackboard or the group coordinator computes the goal satisfaction on the basis of the individual results. Together with the concrete satisfaction level, the goal satisfaction is available for any agent of the group. If the goal-satisfaction is classified as deficient, the agents should adjust their operational autonomy. Doing so, the agents should plan the next action or action sequence under consideration of the global goal. E.g., assume that a BDI agent has instantiated an intention and associates this intention with a partial global plan. The agent would now choose a linearization of the plan which is most suitable for supporting the global goal.

The observation stage is used for normal system performance. If the system performance with respect to a specific goal is critical (or cannot be achieved completely for some time, code “yellow”), the multiagent system’s state changes to the analysis stage. In this stage, the agents have to communicate their currently pursued goals. The analysis is performed by the agents cooperatively or by a central entity (group manager). The group manager has to identify the interdependencies of the goal selections of the individual agents and missing system performance on the group level. These interdependencies are then published. Under consideration of the autonomy of individual agents, the tactical autonomy has to be adjusted by the agents. Each agent should consider the effects of its goal instantiation, e.g., in our example the step of associating a plan to intentions, with respect to the group performance.

There are situations where an uncoordinated treatment of the mismatch of global goals by individual adaptations cannot lead to satisfying results. This can be the case especially if many agents adapt their behavior in a similar way which can lead to the over-achievement of one goal while the performance decreases w.r.t. other system goals. In the case of a severe system performance, the group of agents is transformed into the joint solution group. Here, the group manager mediates the negotiation about individual agents’ goals. The agents are assumed to improve their strategic autonomy, i.e., the agents instantiate those goals which help the group performance.

The solution which has been negotiated in the group and which restored system’s performance is generalized as a social rule for later usage in severe situations (phase four), i.e., the experiences of phase three are made persistent for future situations and costly computation and communication can be avoided by handling similar situations in previous phases. For more details about this approach see [1].

4 Regulated Autonomy

As already mentioned the concept of regulated autonomy is an implementation of the strategy revision, shown in figure 1. It is implemented in as a rule-based approach with a centralized entity for monitoring. Typically for rule-based systems the reaction scheme is statically encoded.

The main idea of the concept of regulated autonomy is sketched in figure 2. In default mode, each agent is free to select its behavior as desired (Fig. 2a). Whenever the system performance reaches a critical state, phase two is initiated. In this phase, the manager agent instructs the shop agents to change their strategy in order to improve the system performance (Fig. 2b). Whenever the strategy is changed, costs of the strategy adaptation is recorded. If the system's overall performance reaches an acceptable status, the entities are allowed to use their own local strategies again (Fig. 2c).

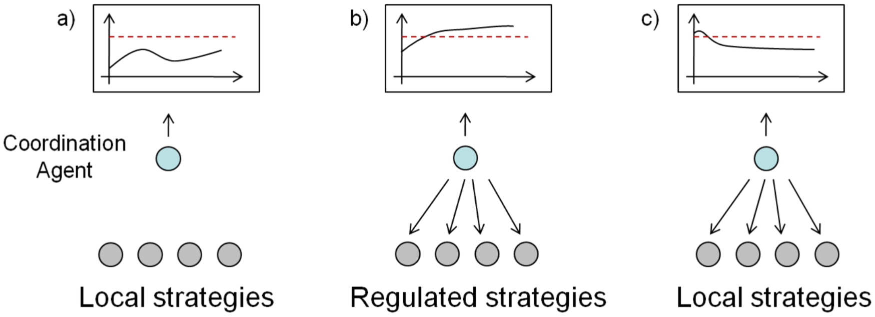


Fig. 2. Autonomy vs. regulation

4.1 Scenario Settings

As in our previous work we use the job shop scenario presented in [32] and [33]. Therefore, we briefly sketch the scenario here. Figure 3 presents a schematic overview of the scenario.

Each shop has an input and an output buffer. It offers exactly one operation. Each job schedules its current jobs using a given dispatching rule. The rules for the shops are assigned randomly from the set of strategies presented in Table 1. These strategies are well known, see e.g. [34].

For simplicity reasons, transportation is not modeled explicitly. It is assumed that enough transport capacity is always available and transportation time is zero. The job characteristics are taken from Brennan and O [33] where the scenario is used as well. In Table 2 the duration and shop sequence are summarized.

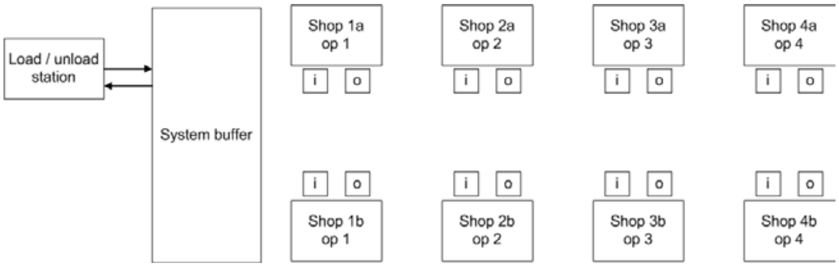


Fig. 3. Shop layout, according to [32]

Strategy Code	Description
SIRO	Service in random order
FIFO	First in first out
SPT	Shortest processing time first
LPT	Longest processing time first
WSPT	Weighted SPT

Table 1. Dispatching strategies for shops

Step / job type	1	2	3	4
J1	6/1	8/2	13/3	5/4
J2	4/1	3/2	8/3	3/4
J3	3/4	6/2	15/1	4/3
J4	5/2	6/1	13/3	4/4
J5	5/1	3/2	8/4	4/3

Table 2. Process plan for different jobs, encoded as time/operation, according to Brennan and O [33]

There exist five different job types which differ in their processing time for each operation and the sequence of operations needed to be performed. The jobs choose the next shop using the shortest queue strategy.

As an objective function for the overall system we use the mean flow time. This implies that, regarding the given dispatching rules, it is known that the SPT dispatching rule will perform best. This eases the application of a rule-based implementation of the strategy revision process. E.g., if another objective function is used to minimize the makespan, the optimal distribution of dispatching rules depends on the set of orders, and has to be computed for each revision process.

As already mentioned there exists one entity, called manager agent, that supervises the overall performance. If a job is finished, the corresponding job agent informs the manager and reports the flow time of this job. The manager agent can monitor the mean flow time according to the jobs finished so far. If this value falls below a specified threshold, the manager agent can order the shop agents to work following a specific strategy. Here this is the SPT strategy which is known to perform best in this scenario. If the actual mean flow time reaches an acceptable range again, it can allow the shops to work according to their locally preferred strategy.

4.2 Evaluation

The results presented in this section were computed using a time driven simulation implemented as a multiagent system based on the JAVA agent development framework JADE[1].

For evaluation purposes, we use three basic settings with different control cycles w.r.t. the overall system's performance. In the `Con01` experiments, the current quality (mean flow time) is checked after each job. In the `Con03` and `Con10` settings, the control interval is set to three and ten, respectively. For each setting, ten different runs are performed where 100 jobs are generated and processed. Figure 4 shows the average mean flow times for all three settings[2]. The mean flow time is computed every time a job has been finished, i.e., the last value (job no. 100) indicates the mean flow time of all 100 jobs.

Figure 5 presents Box-Whisker plots of the relative central control time, i.e., the ratio of time interval lengths under central control divided by the total time. Box-Whisker presents the upper and lower quartile and the median. Therefore, they can be used to discuss the statistically spread of the data. While the mean central control times of setting `Con01` and `Con03` do not differ a lot, the mean value of `Con10` is lower indicating that in our experiments central control is rather infrequent. Having in mind that these regulated strategies are capable to ensure an adequate level of the overall performance (see [5]) it can be stated, that this can be done restricting the local autonomy rarely.

[1] For the Java Agent DEvelopment Framework see `http://jade.tilab.com/`.

[2] All statistical computations as well as plots have been generated with R Project for Statistical Computing 2.6.1, see `http://www.r-project.org/`.

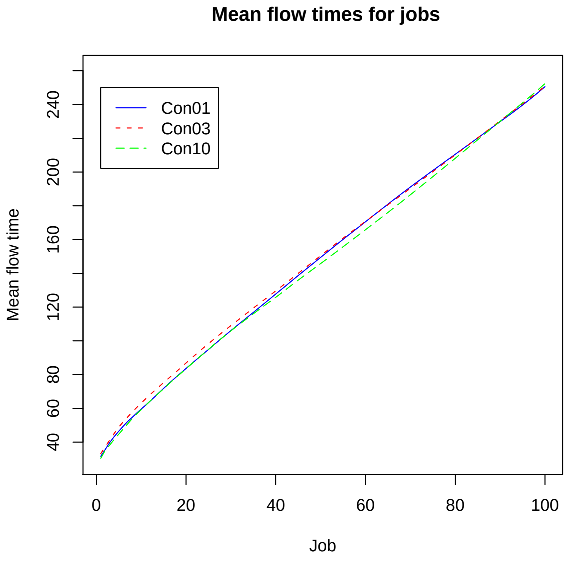


Fig. 4. Mean flow times

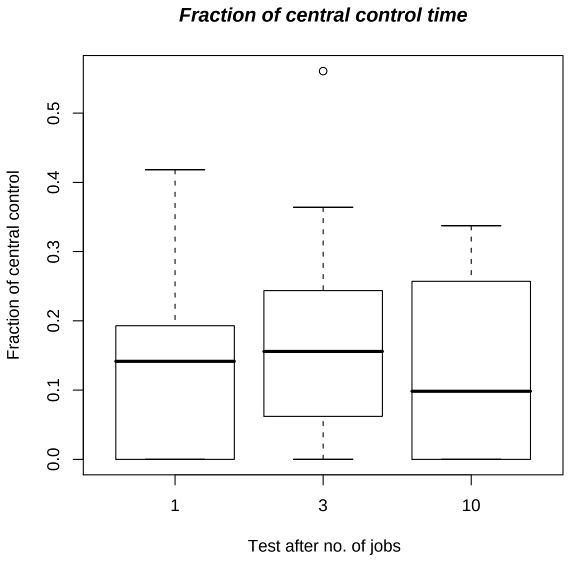


Fig. 5. Box-Whisker plot of relative central control times

5 Reflection and Norms

So far, we presented a methodology for the dynamic adaptation of multiagent systems. The concept itself is – deliberately – on a rather abstract level such that a huge variety of methods for concrete dynamic system design can be derived. Thus, the presented methodology of dynamic adaptation should be rather seen as a building block for software systems; we explicitly do not propose an software architecture. The dynamic adaptation, i.e., the strategic management, of autonomous software systems constitutes the center of our research agenda. Until now we have investigated three aspects. As a first step we[3] have designed the overall methodology as an interdisciplinary model with a mapping of Luhmann's concept of reflection following to strategic management of autonomous software systems. In a second step, we[4] investigated the feasibility of observation and analysis of interdependencies between local and global objectives. In this context, we applied objectives on the basis of key performance indicators and utility functions; the assessment of dependencies are derived automatically. The third aspect, the implementation of the strategy revision process, has been done as a statical rule-based approach (regulated autonomy) which has been described briefly in previous section.

We rely on mechanisms adapted from social science. To stay in line with this research and the terminology, the use of norms seems promising. Having in mind the abstract concept mentioned above, there are no restrictions for the formalism of norms to be applied.

To our point of view, it is promising to use norms for the implementation of the following aspects of the reflection methodology. As already mentioned, the institutionalization phase of the reflection can be implemented as norms to conserve successful strategies persistently.

In our case study (regulated autonomy) we expect high potential of substituting the static rules by notions of norms.

The most challenging process step of our methodology is the generation joint solutions in the recommendation phase. Various approaches like central decision making, argumentation, negotiation are applicable. We are convinced that none of the approaches is dominant with respect to different application domains. Consequently, the process of finding joint solutions should be guided by norms.

Until now the discussion focussed on how to apply norms on the process of reflection. However, we assume that there is high potential for applying our methodology of dynamic adaptation to the evolution of norms. From a more general perspective, this methodology can be used to enrich a model for social simulation.

We are looking forward to discuss these issues in Dagstuhl.

[3] In cooperation with the social scientist Frank Hillebrandt (University of Muenster).
[4] Together with the diploma student Florian Pantke (University of Bremen).

References

1. Timm, I.J., Hillebrandt, F.: Reflexion als sozialer Mechanismus zum strategischen Management autonomer Softwaresysteme. In Schmitt, M., Hillebrandt, F., Florian, M., eds.: Reflexive soziale Mechanismen. VS Verlag für Sozialwissenschaften, Wiesbaden (2006)
2. Schumann, R., Sauer, J.: Implications and consequences of mass customization on manufacturing control. In Blecker, T., Edwards, K., Friedrich, G., Salvador, F., eds.: IMCM'07 + PETO'07. Volume 3 of Series on Business Informatics and Application Systems., Hamburg, GITO (2007) 365 – 378
3. Kirn, S., Herzog, O., Lockemann, P., Spaniol, O.: Multiagent Engineering. International Handbooks on Information Systems. Springer, Berlin et al. (2006)
4. Kephart, J.O., Chess, D.M.: The vision of autonomic computing. Computer **36** (2003) 41–50
5. Schumann, R., Lattner, A.D., Timm, I.J.: Regulated autonomy: A case study. In Mönch, L., Pankratz, G., eds.: Proceedings of the Conference Track "Intelligente Systeme zur Entscheidungsunterstützung" at the Multikonferenz Wirtschaftsinformatik (MKWI 2008), Munich, Germany, SCS Publishing House (2008) 83–98
6. Falcone, R., Castelfranchi, C.: Grounding autonomy adjustment on delegation and trust theory. Journal of Experimental and Theoretical Artificial Intelligence **12** (2000) 149–151
7. Rovatsos, M., Weiss, G.: Autonomous software. In Chang, S.K., ed.: Handbook of Software Engineering and Knowledge Engineering. Volume 3: Recent Advances., River Edge, New Jersey, World Scientific Publishing (2005)
8. Castelfranchi, C., Conte, R.: Emergent functionality among intelligent systems: Cooperation within and without minds. Journal on Artificial Intelligence and Society **6** (1992) 78–87
9. Steenhuisen, J.R., Witteveen, C., ter Mors, A.W., Valk, J.M.: Framework and complexity results for coordinating non-cooperative planning agents. In Fischer, K., Timm, I., André, E., Zhong, N., eds.: Multiagent System Technologies (MATES). LNAI, Erfurt, Springer Verlag, Berlin, Heidelberg, New York (2006) 98 – 109
10. Bertschinger, N., Olbrich, E., Ay, N., Jost, J.: Autonomy: An information-theoretic perspective. Biosystems (2008) 331–345
11. Kirn, S.: Kooperierende intelligente Agenten in Virtuellen Organisationen. HMD **185** (1995) 24 – 36
12. Bradshaw, J.M., Feltovich, P.J., Jung, H., Kuklarni, S., Taysom, W., Uszok, A.: Dimensions of adjustable autonomy and mixed-initiative-interaction. In Nickles, M., Rovatsos, M., Weiss, G., eds.: AUTONOMY 2003 : International Workshop on computational autonomy. Volume 2969 of LNAI., Melbourne, Springer, Berlin (2004) 17 – 39
13. Barber, K.S., Martin, C.E.: Agent autonomy: Specification, measurement, and dynamic adjustment. In: Proceedings of the Autonomy Control Software Workshop at Autonomous Agents 1999 (Agents99), Seattle, WA (1999)
14. Luck, M., d'Inverno, M., Munroe, S.: Autonomy: Variable and generative. In Hexmoor, H., Falcone, R., Castelfranchi, C., eds.: Agent Autonomy. Multiagent Systems, Artificial Societies, and Simulated Organizations. Springer (2003) 9 – 22
15. Schillo, M.: Self-organization and adjustable autonomy: two sides of the same coin? Connection Science **14** (2002) 345–359
16. Schillo, M.: Self-organization and adjustable autonomy: Two sides of the same medal? In Hexmoor, H., Falcone, R., eds.: Proceedings of the AAAI2002 Workshop

on Autonomy, Delegation, and Control: From Inter-agent to Groups, Menlo Park, CA, AAAI Press (2002) 64–71

17. Nickles, M., Rovatsos, M., Weiss, G.: A schema for specifying computational autonomy. In: Proceedings of the Third International Workshop "Engineering Societies in the Agents World" (ESAW'2002), Madrid, Spain (2002)
18. Schurr, N., Marecki, J., Tambe, M.: Riaact: A robust approach to adjustable autonomy for human-multiagent teams. In Padgham, L., Parkes, D.C., Müller, J.P., Parsons, S., eds.: Proc. of 7th Int. Conf. on Autonomous Agents and Multiagent Systems (AAMAS 2008), Estoril, Portugal (2008) 1429–1432
19. Scerri, P., Sycara, K., Tambe, M.: Adjustable autonomy in the context of coordination. In: AIAA 3rd "Unmanned Unlimited" Technical Conference, Workshop and Exhibit. (2004)
20. Urbig, D., Schröter, K.: From full autonomy to manual control. In: Proceedings of the Workshop Concurrency, Specification & Programming (CS&P'05), Ruciae-Nida, Poland (2005) 520–531
21. Gancet, J., Lacroix, S.: Embedding heterogeneous levels of decisional autonomy in multi-robot systems. In: 7th International Symposium on Distributed Autonomous Robotic Systems, Toulouse (France) (2004)
22. Mailler, R.T.: A mediaton-based approach to cooperative, distributed problem solving. PhD thesis, University of Massachusetts Amherst (2004)
23. Barber, K.S., Goel, A., Martin, C.E.: Dynamic adaptive autonomy in multi-agent systems. Journal of Experimental and Theoretical Artificial Intelligence **12** (2000) 129–147
24. Liu, J.: Self-Organized Autonomy in Multi-Agent Systems. In: Autonomous Agents and Multiagent Systems - Explorations in Learning, Self-Organization and Adaptive Computing. World Scientific, Singapore (2001) 141–180
25. Liang, H.H., Zhu, M.L.: Developing self-organized architecture solution according to model driven generative domain engineering. In Czap, H., Unland, R., Branki, C., Tianfield, H., eds.: Self-Organization and Autonomic Informatics (I). Volume 135 of Frontiers in Artificial Intelligence and Application., Amsterdam, IOS Press (2005) 97–104
26. de Wolf, T., Holvoet, T.: Towards a methodology for engineering self-organising emergent systems. In Czap, H., Unland, R., Branki, C., Tianfield, H., eds.: Self-Organization and Autonomic Informatics (I). Volume 135 of Frontiers in Artificial Intelligence and Application., Amsterdam, IOS Press (2005) 18–34
27. Schegloff, E.A.: Between macro and micro: Contexts and other connections. In Alexander, J., Giesen, B., Münch, R., eds.: The Micro-Macro Link. University of California Press, Berkeley (1987) 207–237
28. Bohman, J.: The Macro-Micro Relation. In: New Philosophy of Social Science - Problems of Indeterminancy. The MIT Press, Cambridge (1993) 146–185
29. Alexander, J.C., Giesen, B.: From reduction to linkage: The long view of the micro-macro link. In Alexander, J., Giesen, B., Münch, R., eds.: The Micro-Macro Link. University of California Press, Berkeley (1987) 1–45
30. Luhmann, N.: Soziale Systeme, Grundrisse einer allgemeinen Theorie. Suhrkamp, Frankfurt (1984)
31. Sandholm, T.: Distributed rational decision making. In Weiss, G., ed.: Multiagent Systems: a modern approach to distributed artificial intelligence. MIT Press (1999) 201– 258
32. Cavalieri, S., Bongaerts, L., Macchi, M., Taisch, M., Weyns, J.: A benchmark framework for manufacturing control. In: Second International Workshop on Intelligent Manufacturing Systems, Leuven, Belgium (1999) 225 – 236

33. Brennan, R.W., O, W.: A simulation test-bed to evaluate multi-agent control of manufacturing systems. In: WSC '00: Proceedings of the 32nd Conference on Winter Simulation, Orlando, Florida, Society for Computer Simulation International (2000) 1747–1756
34. Pinedo, M.: Scheduling: Theory, Algorithms and Systems. Industrial and Systems Engineering. Prentice-Hall (1995)

Robust Normative Systems[⋆]

Thomas Ågotnes[1], Wiebe van der Hoek[2], and Michael Wooldridge[2]

[1] Dept. of Computer Engineering, Bergen University College, Norway
[2] Dept. of Computer Science, University of Liverpool, UK

Abstract. Although normative systems, or social laws, have proved to be a highly influential approach to coordination in multi-agent systems, the issue of *compliance* to such normative systems remains problematic. In all real systems, it is possible that some members of an agent population will not comply with the rules of a normative system, even if it is in their interests to do so. It is therefore important to consider the extent to which a normative system is *robust*, i.e., the extent to which it remains effective even if some agents do not comply with it. We formalise and investigate three different notions of robustness and related decision problems. We begin by considering sets of agents whose compliance is necessary and/or sufficient to guarantee the effectiveness of a normative system; we then consider quantitative approaches to robustness, where we try to identify the proportion of an agent population that must comply in order to ensure success, and finally, we consider a more general approach, where we characterise the compliance conditions required for success as a logical formula.

1 Introduction

Normative systems, or social laws, have been widely promoted as an approach to coordinating multi-agent systems [11, 12, 6, 8, 1, 2]. The basic idea is that a normative system is a set of constraints on the behaviour of agents in the system; after imposing these constraints, it is intended that some desirable overall property will hold. One of the most important issues associated with such normative systems – and one of the most ignored – is that of *compliance*. Put simply, what happens if some system participants do not comply with the regulations of the normative system? Non-compliance may be accidental (e.g., a message fails and so some participants are not informed about the regulations). Alternatively, it may be deliberate but rational (e.g., a participant chooses to ignore the norms because it does not see them as being in its own best interests), or deliberately irrational (e.g., a computer virus). Whatever the cause, it seems inevitable that, in real, large-scale systems, non-compliance will occur, and it is therefore important to consider the consequences of non-compliance. Existing research has addressed the issue of non-compliance in at least two ways.

First, one can design the normative system taking the goals and aspirations of system participants into account, so that compliance is the rational choice for participants [2]. Using the terminology of mechanism design [10, p.179], we try to make

[⋆] The content of this paper is also found in a paper appearing in the proceedings of the AAMAS 2008 conference.

Dagstuhl Seminar Proceedings 09121
Normative Multi-Agent Systems
http://drops.dagstuhl.de/opus/volltexte/2009/1897

compliance *incentive compatible*. Where this approach is available, it seems highly attractive. However, given some desired objective for a normative system, it is not always possible to construct an incentive compatible normative system that achieves some outcome, and even where it is possible, it is still likely that large, open systems will fall prey to irrational behaviour.

Second, one can combine the normative system with some *penalty* mechanism, to punish non-compliance [4]. The advantage of this approach is that it can be applied to most scenarios, and that it is familiar (this is, after all, how normative systems often work in the real world). There are many disadvantages, however. For example, it may be hard to detect when non-compliance has occurred, and in large, Internet-like systems, it may be hard to impose penalties (e.g., across national borders).

For these reasons, in this paper we introduce the notion of *robustness* for normative systems. Intuitively, a normative system is robust to the extent to which it remains effective in the event of non-compliance by some agents. Following an introduction to the technical framework of normative systems, we introduce and investigate three ways of characterising robustness. First, we consider trying to identify coalitions whose compliance is *necessary* and/or *sufficient* to ensure that the normative system is effective. We characterise the complexity of checking these notions of robustness, and consider cases where verifying these notions of robustness is easier. In addition to verification we consider the complexity of *robust feasibility* of a normative system: given a reliable coalition, does there exist a normative system which is effective whenever that coalition complies? We then consider a more *quantitative* notion of robustness, called k*-robustness*, where we try to identify the *number* of agents that could deviate and still leave the normative system effective. Finally, we consider a more general, *logical* approach of characterising robustness, whereby we define a predicate over sets of agents, such that this predicate characterises exactly those sets of agents whose compliance will ensure the success of the normative system. We conclude with a brief discussion, including some pointers to related and future work.

2 Formal Preliminaries

In this section, we present the formal framework for normative systems that we use throughout the remainder of the paper. This framework is based on that of [8, 1, 2], which is in turn descended from [11]. Although our presentation is complete, it is succinct, and readers are referred to [8, 1, 2] for details and discussion.

Kripke Structures: We use *Kripke structures* as our basic semantic model for multi-agent systems [5]. A Kripke structure is essentially a directed graph, with the vertex set S corresponding to possible *states* of the system being modelled, and the relation $R \subseteq S \times S$ capturing the possible *transitions* of the system; $S^0 \subseteq S$ denotes the *initial states* of the system. Intuitively, transitions are caused by *agents* in the system performing *actions*, although we do not include such actions in our semantic model (see, e.g., [11, 8] for models which include actions as first class citizens). An arc $(s, s') \in R$ corresponds to the execution of an atomic action by one of the agents in the system. Note that we are therefore here *not* modelling *synchronous* action. This assumption is not essential, but it simplifies the presentation. However, we find it convenient to include within our model the agents that cause transitions. We therefore assume a set A of agents, and we label each transition in R with the agent that causes the transition via a function $\alpha : R \rightarrow A$. Finally, we use a vocabulary $\Phi = \{p, q, \ldots\}$ of Boolean variables to express the properties of individual states S: we use a function $V : S \rightarrow 2^{\Phi}$ to label each state with the Boolean variables true (or satisfied) in that state.

Formally, an *agent-labelled Kripke structure* (over Φ) is a 6-tuple:

$$K = \langle S, S^0, R, A, \alpha,\ V \rangle,$$

where: S is a finite, non-empty set of *states*; $S^0 \subseteq S$ ($S^0 \neq \emptyset$) is the set of *initial states*; $R \subseteq S \times S$ is a total binary relation on S, which we refer to as the *transition relation*; $A = \{1, \ldots, n\}$ is a set of *agents*; $\alpha : R \rightarrow A$ labels each transition in R with an agent; and $V : S \rightarrow 2^{\Phi}$ labels each state with the set of propositional variables true in that state.

We hereafter refer to an agent-labelled Kripke structure simply as a *Kripke structure*. A *path* over a transition relation R is an infinite sequence of states $\pi = s_0, s_1, \ldots$ such that $\forall u \in \mathbb{N}$: $(s_u, s_{u+1}) \in R$. If $u \in \mathbb{N}$, then we denote by $\pi[u]$ the component indexed by u in π (thus $\pi[0]$ denotes the first element, $\pi[1]$ the second, and so on). A path π such that $\pi[0] = s$ is an *s-path*. Let $\Pi_R(s)$ denote the set of s-paths over R; since it will usually be clear from context, we often omit reference to R, and simply write $\Pi(s)$. We will sometimes refer to and think of an s-path as a possible computation, or system evolution, from s.

CTL: We use Computation Tree Logic (CTL), a well-known and widely used branching time temporal logic, to express the *objectives* of normative systems [5]. Given a set $\Phi = \{p, q, \ldots\}$ of atomic propositions, the syntax of CTL is defined by the following grammar, where $p \in \Phi$:

$$\varphi ::= \top \mid p \mid \neg\varphi \mid \varphi \vee \varphi \mid \mathsf{E}\bigcirc\varphi \mid \mathsf{E}(\varphi\,\mathcal{U}\,\varphi) \mid \mathsf{A}\bigcirc\varphi \mid \mathsf{A}(\varphi\,\mathcal{U}\,\varphi)$$

The semantics of CTL are given with respect to the satisfaction relation "$\models$", which holds between *pointed structures* K, s, (where K is a Kripke structure and s is a state in K), and formulae of the language. The satisfaction relation is defined as follows:

$K, s \models \top$;
$K, s \models p$ iff $p \in V(s)$ (where $p \in \Phi$);
$K, s \models \neg\varphi$ iff not $K, s \models \varphi$;
$K, s \models \varphi \vee \psi$ iff $K, s \models \varphi$ or $K, s \models \psi$;
$K, s \models \mathsf{A}\bigcirc\varphi$ iff $\forall \pi \in \Pi(s) : K, \pi[1] \models \varphi$;
$K, s \models \mathsf{E}\bigcirc\varphi$ iff $\exists \pi \in \Pi(s) : K, \pi[1] \models \varphi$;
$K, s \models \mathsf{A}(\varphi\,\mathcal{U}\,\psi)$ iff $\forall \pi \in \Pi(s), \exists u \in \mathbb{N}$, s.t. $K, \pi[u] \models \psi$ and $\forall v, (0 \leq v < u) : K, \pi[v] \models \varphi$
$K, s \models \mathsf{E}(\varphi\,\mathcal{U}\,\psi)$ iff $\exists \pi \in \Pi(s), \exists u \in \mathbb{N}$, s.t. $K, \pi[u] \models \psi$ and $\forall v, (0 \leq v < u) : K, \pi[v] \models \varphi$

The remaining classical logic connectives ("$\wedge$", "$\rightarrow$", "$\leftrightarrow$") are defined as abbreviations in terms of $\neg, \vee$ in the conventional way. The remaining CTL temporal operators are defined:

$$\mathsf{A}\Diamond\varphi \equiv \mathsf{A}(\top\,\mathcal{U}\,\varphi) \qquad \mathsf{E}\Diamond\varphi \equiv \mathsf{E}(\top\,\mathcal{U}\,\varphi)$$
$$\mathsf{A}\,\Box\varphi \equiv \neg\mathsf{E}\Diamond\neg\varphi \qquad \mathsf{E}\,\Box\varphi \equiv \neg\mathsf{A}\Diamond\neg\varphi$$

We say φ is *satisfiable* if $K, s \models \varphi$ for some Kripke structure K and state s in K; φ is *valid* if $K, s \models \varphi$ for all Kripke structures K and states s in K. The problem of checking whether $K, s \models \varphi$ for given K, s, φ (*model checking*) can be done in deterministic polynomial time, while checking whether a given φ is satisfiable or whether φ is valid is EXPTIME-complete [5]. We write $K \models \varphi$ if $K, s_0 \models \varphi$ for all $s_0 \in S^0$, and $\models \varphi$ if $K \models \varphi$ for all K.

Later, we will make use of two fragments of CTL: the universal language L^u (with typical element μ), and the existential fragment L^e (typical element ε):

$$\mu ::= \top \mid \bot \mid\ p \mid \neg p \mid \mu \vee \mu \mid \mu \wedge \mu \mid \mathsf{A}\bigcirc\mu \mid \mathsf{A}\,\Box\mu \mid \mathsf{A}(\mu\,\mathcal{U}\,\mu)$$
$$\varepsilon ::= \top \mid \bot \mid\ p \mid \neg p \mid \varepsilon \vee \varepsilon \mid \varepsilon \wedge \varepsilon \mid \mathsf{E}\bigcirc\varepsilon \mid \mathsf{E}\,\Box\varepsilon \mid \mathsf{E}(\varepsilon\,\mathcal{U}\,\varepsilon)$$

The key point about these fragments is as follows. Let us say, for two Kripke structures $K_1 = \langle S, S^0, R_1, A, \alpha,\ V\rangle$ and $K_2 = \langle S, S^0, R_2, A, \alpha,\ V\rangle$ that K_1 is a subsystem of K_2 and K_2 is a supersystem of K_1, (denoted $K_1 \sqsubseteq K_2$), iff $R_1 \subseteq R_2$. Then we have (cf. [8]).

Theorem 1 ([8]). *Suppose $K_1 \sqsubseteq K_2$, and $s \in S$. Then:*

$$\begin{array}{lcl} \forall\varepsilon \in L^e : K_1, s \models \varepsilon & \Rightarrow & K_2, s \models \varepsilon; \quad \textit{and} \\ \forall\mu \in L^u : K_2, s \models \mu & \Rightarrow & K_1, s \models \mu. \end{array}$$

Normative Systems: For our purposes, a *normative system* (or "norm") is simply *a set of constraints on the behaviour of agents in a system* [1]. More precisely, a normative system defines, for every possible system transition, whether or not that transition is considered to be legal or not. Different normative systems may differ on whether or not a transition is legal. Formally, a normative system η (w.r.t. a Kripke structure $K = \langle S, S^0, R, A, \alpha,\ V\rangle$) is simply a subset of R, such that $R \setminus \eta$ is a total relation. The requirement that $R \setminus \eta$ is total is a *reasonableness* constraint: it prevents normative systems which lead to states with no successor. Let $N(R) = \{\eta : (\eta \subseteq R)\ \&\ (R \setminus \eta \text{ is total})\}$ be the set of normative systems over R. The intended interpretation of a normative system η is that $(s, s') \in \eta$ means transition (s, s') is forbidden in the context of η. We denote the *empty* normative system by $\eta_\emptyset$, i.e., $\eta_\emptyset = \emptyset$. Let $A(\eta) = \{\alpha(s, s') \mid (s, s') \in \eta\}$ denote the set of agents involved in η.

The effect of *implementing* a normative system on a Kripke structure is to eliminate from it all transitions that are forbidden according to this normative system (see [8, 1]). If K is a Kripke structure, and η is a normative system over K, then $K \dagger \eta$ denotes the Kripke structure obtained from K by deleting transitions forbidden in η. Formally, if $K = \langle S, S^0, R, A, \alpha,\ V\rangle$, and $\eta \in N(R)$, then let $K \dagger \eta = K'$ be the Kripke structure $K' = \langle S', S^{0'}, R', A', \alpha',\ V'\rangle$ where:

- $S = S'$, $S^0 = S^{0'}$, $A = A'$, and $V = V'$;
- $R' = R \setminus \eta$; and

– α' is the restriction of α to R':

$$\alpha'(s, s') = \begin{cases} \alpha(s, s') & \text{if } (s, s') \in R' \\ \text{undefined otherwise.} \end{cases}$$

The next most basic question we can ask in the context of normative systems is as follows. We are given a Kripke structure K, representing the state transition graph of our system, and we are given a CTL formula φ, representing the *objective* of a normative system designer (that is, the objective characterises what a designer wishes to accomplish with a normative system). The *feasibility* problem is then whether or not there exists a normative system η such that implementing η in K will achieve φ, i.e., whether $K \dagger \eta \models \varphi$. We say that η is effective for φ in K if $K \dagger \eta \models \varphi$.

We make use of operators on normative systems which correspond to groups of agents "defecting" from the normative system. Formally, let $K = \langle S, S^0, R, A,\alpha,\ V\rangle$ be a Kripke structure, let $C \subseteq A$ be a set of agents over K, and let η be a normative system over K. Then $\eta \upharpoonright C$ denotes the normative system that is the same as η except that it only contains the arcs of η that correspond to the actions of agents in C, i.e., $\eta \upharpoonright C = \{(s, s') : (s, s') \in \eta$ & $\alpha(s, s') \in C\}$. Also, $\eta \upharpoonleft C$ denotes the normative system that is the same as η except that it only contains the arcs of η that *do not* correspond to actions of agents in C: $\eta \upharpoonleft C = \{(s, s') : (s, s') \in \eta$ & $\alpha(s, s') \notin C\}$.

3 Necessity and Sufficiency

As we noted in the introduction, the basic intuition behind robust normative systems is that they remain effective in the presence of deviation, or non-compliance, by some members of the agent population. As we shall see, there are several different ways of formulating robustness. Our first approach is to try to characterise "lynchpin" agents – those agents whose compliance with the normative system is somehow crucial for the successful operation of the system. This seems appropriate when there are "key players" in the normative system – for example, where there is a single point of failure. In this section, we therefore consider coalitions whose compliance is *necessary and/or sufficient* to ensure that the normative system is effective.

We say that $C \subseteq A$ are *sufficient* for η in the context of K and φ if the compliance of C with η is effective, i.e., iff:

$$\forall C' \subseteq A : (C \subseteq C') \quad \Rightarrow \quad [K \dagger (\eta \upharpoonright C') \models \varphi].$$

The following example illustrates this notion of sufficiency.

Example 1. Consider four agents who are attending a conference with an on-site computer facility. This service centre has currently one printer, two scanners and three PCs available. Agent a has tasks that require access to a printer and PC, agent b needs a printer and scanner, agent c is in need of a scanner and PC and agent d will need a scanner only. The set of agents is $A = \{a, b, c, d\}$. They are interested in using resources of type R_1, R_2, R_3, of each resource type R_j there are j instances of each: $R_1 = \{printer_1\}$, $R_2 = \{scanner_1, scanner_2\}$, $R_3 = \{pc_1, pc_2, pc_3\}$. At a given point in time, a resource can be owned by an agent. The actions available to the agents are making available a resource they currently own, or taking possession of a resource which is available. We assume that the agents never act at exactly the same time; in particular we assume that actions are turn-based – first a can perform some action, then b, and so on. A state s is a tuple

$$s = \langle O_a, O_b, O_c, O_d, i\rangle$$

where, for each $i \in A$, O_i is the set of resources currently owned by i.

The number of agents that own a resource of type j cannot be greater than j. Let, for each resource R_j and state s, $avail(j, s)$ be the number of resources of type j that are not owned by an agent. The component $i \in A$ of s denotes whose turn it is: we write $turn(s) = i$. If $R_j \cap O_i \neq \emptyset$, we say that i owns a resource of type j and write $R_j \prec O_i$.

Our agents are not equal. In order to fullfil his task, agent a would every now and then like to use resources of type R_1 and R_3 simultaneously. We write $\mathit{Useful}(a) = \{R_1, R_3\}$. Simililary, $\mathit{Useful}(b) = \{R_1, R_2\}$, $\mathit{Useful}(c) = \{R_2, R_3\}$ while $\mathit{Useful}(d) = \{R_2\}$.

Let $s = \langle O_a, O_b, O_c, O_d, i\rangle$ and $s' = \langle O'_a, O'_b, O'_c, O'_d, i'\rangle$ be two states. Then $(s, s') \in R$ iff

1. $a' = b, b' = c, c' = d$ and $d' = a$;
2. for all $k \neq i$ and all j: $R_j \prec O_k \Leftrightarrow R_j \prec O'_k$;
3. if $R_j \prec O'_i$ and $R_j \not\prec O_i$ then $avail(j, s) > 0$.

Furthermore, $\alpha(s, s') = i$ when $turn(s) = i$.

Let the starting state of the system be such that it is agent a's turn, and nobody owns any resource. If we call this system K_0, then a first norm η_0 we impose on K is that no agent (i) owns two resources of the same type at the same time, (ii) takes posession of a resource that he does not need, (iii) takes possession of two new resources simultaneously, and (iv) fails to take possession of some useful resource if it is available when it is his turn:

$$\eta_0 = \left\{ (s, s') \mid \begin{array}{l} turn(s) = i, \text{ and} \\ (\exists j : |O'_i \cap R_j| \geq 2, \text{ or} \\ \exists j : |O'_i \cap R_j| \geq 1 \text{ and } R_j \notin \mathit{Useful}(i), \text{ or} \\ \exists x, y : x \neq y, x, y \in O'_i \text{ and } x, y \notin O_i, \text{ or} \\ \forall j : (R_j \in \mathit{Useful}(i), |O_i \cap R_j| = 0, \\ \quad avail(j, s) > 0) \;\Rightarrow |O'_i \cap R_j| = 0). \end{array} \right\}$$

Let $K_1 = K_0 \dagger \eta_0$. Now, in order to formulate some objectives of the system, let a^o_j denote that agent a owns a resource of type j and similarly for the other agents. Let

$$happy(i) = \bigwedge_{R_j \in \mathit{Useful}(i)} i^o_j$$

Thus $happy(i)$ means that i is in possession of all his useful resources, simultaneously. Our first objective is:

$$\varphi_1 = \mathsf{A}\Box \bigwedge_{i \in A} \mathsf{A}\Diamond happy(i).$$

The normative system that we will use for it is

$$\eta_1 = \{(s, s') \mid \mathit{turn}(s) = i \ \& \ O_i = \mathit{Useful}(i) \& \ O_i' \neq \emptyset\}$$

In words: if at some point an agent simultaneously owns all the resources that are useful for him, then he will make them available if it is his turn. Which coalitions are sufficient for this norm in the context of K_1 and φ_1? First of all, consider a coalition without agent a. If a does not comply with norm η_1, then he can grab the printer and hold on to it forever. Thus, agent b will not be happy, because there is only one printer. The same argument holds for a coalition without agent b. Thus, it seems that any sufficient coalition must include both agents a and b. But $\{a, b\}$ alone is not a sufficient coalition, as the following scenario illustrates: (1) a grabs a PC; (2) b grabs the printer; (3) c grabs a scanner; (4) d grabs the other scanner. Now, if c and d do not comply with η_1, it might be that they never give up their scanners, in which case b never will be happy. However, if a and b are joined by c in complying with η_1, the objective is obtained:

$$K_1 \dagger (\eta_1 \upharpoonright \{a, b, c\}) \models \varphi_1$$

– it is easy to see that in fact $\{a, b, c\}$ is sufficient for η_1 in the context of K_1 and φ_1. But $\{a, b, c\}$ and its extension $\{a, b, c, d\}$ are not the only sufficient coalitions in this context: $\{a, b, d\}$ is also sufficient.

Now, associated with this notion is a decision problem: we are given K, η, φ, and C, and asked whether C are sufficient for η in the context of K and φ. It may appear at first sight that this is an easy decision problem: don't we just need to check that $K \dagger (\eta \upharpoonright C) \models \varphi$? The answer is no. For suppose the objective is an *existential* property $\eta \in L^e$. Then the fact that $K \dagger (\eta \upharpoonright C) \models \eta$ and $C \subseteq C'$ does not guarantee that $K \dagger (\eta \upharpoonright C') \models \eta$. Intuitively, this is because, if more agents than C comply, then this might eliminate transitions from K, causing the existential property η to be falsified.

Example 2. We continue Example 1. To demonstrate that sufficiency for a norm in the context of a system and an objective is not monotonic in the coalition C, consider the following existential objective:

$$\varphi_2 = \mathsf{E}\square\neg \mathit{happy}(b)$$

That is, it is possible that b is forever unhappy (we will not discuss *why* the designer of the normative system might have such an objective). We have that:

$$K_1 \dagger (\eta_1 \upharpoonright \{b\}) \models \varphi_2.$$

That is, if b complies with the norm η_1, the objective is true. This is because, for example, agent a can block b's access to the printer. However, as we saw in Example 1, $K_1 \dagger (\eta_1 \upharpoonright \{a, b, c\}) \models \neg\varphi_2$, so $\{b\}$ is not sufficient for the objective φ_2.

We can prove that, in general, checking sufficiency is computationally hard.

Theorem 2. *Deciding C-sufficiency is co-*NP*-complete.*

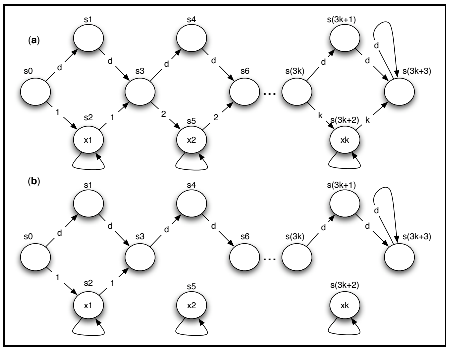


Fig. 1. Illustrating the reduction used in Theorem 2: (a) the Kripke structure produced in the reduction; (b) how the construction corresponds to a valuation: if only agent 1 defects, then the Kripke structure we obtain corresponds to a valuation in which x_1 is true (a state in which x_1 is true is reachable in the resulting structure – $\mathsf{E}\Diamond x_1$ in the objective we construct) and all other variables are false (i.e., are true in unreachable states).

Proof. Membership of co-NP is straightforward from the definitions of the problems. We prove hardness by reducing TAUT, the problem of showing that a formula Ψ of propositional logic is a tautology, i.e., is true under all interpretations. Let $x_1, \ldots, x_k$ be the Boolean variables of Ψ. The reduction is as follows. For each Boolean variable x_i we create an agent a_i, and in addition create one further agent, d. We create $3k + 3$ states, and create the transition relation R and associated agent labelling α and valuation V as illustrated in Figure 1(a): inside states are the propositions true in that state, while arcs between states are labelled with the agent associated with the transition. Let $S^0 = \{s_0\}$ be the singleton initial state set. We have thus defined the Kripke structure K. For the remaining components, define $C = \emptyset$, $\eta = \{(s_0, s_2), (s_2, s_3), (s_3, s_5), (s_5, s_6), \ldots, (s_{3k+2}, s_{3k+3})\}$ (i.e., all the lower arcs in the figure), and finally, define φ to be the formula obtained from Ψ by systematically replacing each Boolean variable x_i by $(\mathsf{E}\Diamond x_i)$. Now, we claim that η is C-sufficient for φ in K iff Ψ is a tautology. First, notice that since $C = \emptyset$, then for all $C' \subseteq A$, we have $C \subseteq C'$, and so the problem reduces to the following:

$$\forall C' \subseteq A : [K \dagger (\eta \restriction C') \models \varphi].$$

The correctness of the reduction is illustrated in Figure 1(b), where we show the Kripke structure obtained when only agent 1 defects from the normative system; in this case, the Kripke structure we obtain corresponds to a valuation of Ψ which makes variable x_1 true and all others false.

However, the news is not all bad: for *universal* objectives, checking sufficiency is easy.

Corollary 1. *Deciding C-sufficiency for objectives $\mu \in L^u$ is polynomial time decidable.*

Proof. Simply check that $K \dagger (\eta \upharpoonright C) \models \mu$; since $\mu \in L^u$, the fact that $K \dagger (\eta \upharpoonright C') \models \mu$ for all $C \subseteq C' \subseteq A$ follows from Theorem 1.

Next, we consider the obvious counterpart notion to sufficiency; that of *necessity*. We say that C are *necessary* for η in the context of K and φ iff C *must* comply with η in order for it to be effective, i.e., iff:

$$\forall C' \subseteq A : [K \dagger (\eta \upharpoonright C') \models \varphi] \quad \Rightarrow \quad (C \subseteq C').$$

The following example illustrates necessity.

Example 3. We continue Example 1. We observed that $\{a, b, c\}$ and $\{a, b, d\}$ are sufficient for η_1 in the context of K_1 and φ_1. Indeed, $\{a, b\}$ is necessary for η_1 in the context of K_1 and φ_1. Both a and b *must* comply with the norm for the objective to be satisfied.

Theorem 3. *Deciding C-necessity is co-*NP*-complete.*

Proof. Membership of co-NP is obvious from the statement of the problem, so consider hardness. Note that proof of Theorem 2 does not go through for this case: since we set $C = \emptyset$ in the reduction, C are trivially necessary. However, we can use the same basic construction as Theorem 2 to prove NP-hardness of the complement problem to C-necessity, i.e., the problem of showing that

$$\exists C' \subseteq A : [K \dagger (\eta \upharpoonright C) \models \varphi] \wedge \neg(C \subseteq C').$$

We reduce SAT. Given a SAT instance Ψ, we follow the construction of Theorem 2, except that set the input coalition C to be $C = \{d\}$. It is now easy to see, using a similar argument to Theorem 2, that Ψ is satisfiable iff $\exists C \subseteq A : [K \dagger (\eta \upharpoonright C) \models \varphi] \wedge \neg(C \subseteq C')$.

The following sums up some general properties of the concepts we have discussed so far. Here, "sufficient" ("necessary") means "sufficient (necessary) for η in the context of K and φ".

Proposition 1.

1. *There might be no sufficient coalitions.*
2. *There is always a necessary coalition: the empty coalition.*
3. *There might be two disjoint sufficient coalitions.*
4. *There might be no non-empty necessary coalitions.*
5. *If C is necessary and C' sufficient, then $C \subseteq C'$.*
6. *If there are two disjoint sufficient coalitions, then there is no non-empty necessary coalition.*

Proof.

1. Trivial. Take, e.g., a system consisting of a single state with a self-loop and where p is true, and let $\varphi = \mathsf{E}\bigcirc\neg p$. η must be empty, and φ can never be true.
2. Immediate.

3. Take again the system from the first point, and let $\varphi = \mathsf{E}\bigcirc p$. Both $\{a\}$ and $\{b\}$ are sufficient, for any $a \neq b$.
4. Take the system and formula in the previous point.
5. Let C be necessary and C' sufficient. From sufficiency of C' we have that $K \dagger (\eta \upharpoonright C') \models \varphi$, and from necessity of C it follows that $C \subseteq C'$.
6. Immediate from the above point.

Note that point 5 above implies that every necessary coalition is contained in the intersection of all sufficient coalitions. Does the other direction hold, i.e., is the intersection of all sufficient coalitions necessary? In the general case the answer is "no" , as the following example illustrates.

Example 4. Take the system in Figure 2, and let $\varphi = \mathsf{E}\bigcirc\mathsf{A}\bigcirc p$. It is easy to see that:

- $\{a\}$ is sufficient;
- $K \dagger (\eta \upharpoonright \{b\}) \models \varphi$;
- None of $\{b\}$, $\{c\}$ or $\{b, c\}$ are sufficient.

From the first and last point it follows that $\{a\}$ is the intersection of all sufficent coalitions; from the second point it follows that $\{a\}$ is not necessary.

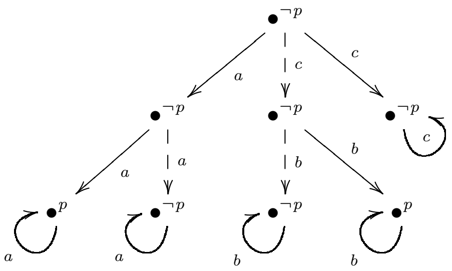


Fig. 2. A normative system. The dashed lines indicate "illegal" transitions. The uppermost state is the single inital state.

However, for universal objectives the greatest necessary coalition is exactly the intersection of the sufficient coalitions:

Lemma 1. *When the objective is a formula in L^u, the intersection of all sufficient coalitions is a necessary coalition.*

Proof. Let $\varphi \in L^u$ and let $C = \bigcap_{C' \text{ sufficient}} C'$. Assume that $K \dagger (\eta \upharpoonright C_2) \models \varphi$; we must show that $C \subseteq C_2$. From Theorem 1 we have $K \dagger (\eta \upharpoonright C_3) \models \varphi$ for any C_3 such that $C_2 \subseteq C_3$. It follows that C_2 is sufficient. But then $C \subseteq C_2$.

Thus, for the case of universal objectives the necessary coalitions are exactly the subsets of the intersection of the sufficient coalitions. Indeed, in Examples 1 we saw that the intersection of the sufficient coalitions, consisting of agents a and b, is a necessary coalition.

3.1 Feasibility of Robust Normative Systems

So far, our technical results have focussed on *verifying* robustness properties of normative systems. However, an equally important question is that of *feasibility*. As we noted earlier, feasibility basically asks whether there exists some normative system such that, if this law was imposed (and, implicitly, everybody complies), then the desired effect of the normative system would be achieved. In the context of robustness, we ask whether a normative system is *robustly* feasible. In more detail, we can think about robust feasibility as follows. Suppose we know that some subset C of the overall agent population is "reliable", in that we are confident that C can be relied upon to comply with a normative system. Then instead of asking whether there exists an *arbitrary* normative system η that is effective for our desired objective φ, we can ask whether there exists a normative system η such that C is sufficient for η in the context of φ. We call this property *C-sufficient feasibility*[3]. Formally, this question is as follows:

$$\begin{aligned}&\exists \eta \in N(R) : (K \dagger \eta \models \varphi) \wedge \\ &\quad \forall C' \subseteq A : (C \subseteq C') \Rightarrow [K \dagger (\eta \upharpoonright C') \models \varphi].\end{aligned}$$

It turns out that, under standard complexity theoretic assumptions, checking this property is harder than the (co-NP-complete) verification problem.

Theorem 4. *Deciding C-sufficient feasibility is Σ_2^p-complete.*

Proof. We deal with the complement of the problem, which we show to be Π_2^p-complete. The complement problem is that of deciding:

$$\begin{aligned}&\forall \eta \in N(R) : (K \dagger \eta \models \varphi) \Rightarrow \\ &\qquad \exists C' \subseteq A : (C \subseteq C') \wedge (K \dagger (\eta \upharpoonright C') \not\models \varphi).\end{aligned}$$

Membership is immediate from the definition of the problem. For hardness, we reduce the problem of determining whether QBF$_{2,\forall}$ formulae are true [9, p.96]. An instance of QBF$_{2,\forall}$ is given by a quantified Boolean formula with the following structure:

$$\forall \bar{x}_1 \, \exists \bar{x}_2 \, \chi(\bar{x}_1, \bar{x}_2) \tag{1}$$

in which $\bar{x}_1$ and $\bar{x}_2$ are disjoint sets of Boolean variables, and $\chi(\bar{x}_1, \bar{x}_2)$ is a propositional logic formula (the *matrix*) over these variables. Such a formula is true if for all assignments to Boolean variables $\bar{x}_1$, there exists an assignment to $\bar{x}_2$, such that $\chi(\bar{x}_1, \bar{x}_2)$ is true under the overall assignment. An example of a QBF$_{2,\forall}$ formula is:

$$\forall x_1 \exists x_2 [(x_1 \vee x_2) \wedge (x_1 \vee \neg x_2)] \tag{2}$$

The reduction is related to that of Theorem 2, although slightly more involved. Let $\bar{x} = \{x_1, \ldots, x_g\}$ be the universally quantified variables in the input formula, let $\bar{y} =$

[3] It may at first sight seem strange that we consider this problem: why not simply look for a normative system η such that $A(\eta) = C$? Our rationale is that the *worst case* corresponds to only C complying with the normative system; it may well be that we get *better* results if more agents comply.

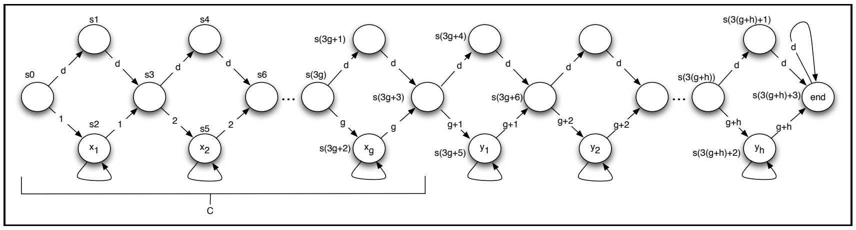


Fig. 3. Illustrating the reduction used in Theorem 4.

$\{y_1, \ldots, y_h\}$ be the existentially quantified variables, and let $\chi(\bar{x}, \bar{y})$ be the matrix. We create a Kripke structure with $3(3(g + h) + 3)$ states and $g + h$ agents. We create variables corresponding to $\bar{x}$ and $\bar{y}$, and in addition to these, we create a variable end. The overall structure is defined to be as shown in Figure 3; note that end is true only in the final state of the structure. We set $C = \{1, \ldots, g\}$, and create the objective φ to be

$$\varphi \hat{=} (\neg \mathsf{E} \Diamond end) \vee (\neg \chi^*(\bar{x}, \bar{y}))$$

where $\chi^*(\bar{x}, \bar{y})$ is the CTL formula obtained from the propositional formula $\chi(\bar{x}, \bar{y})$ by systematically substituting $(\mathsf{E} \Diamond v)$ for each variable $v \in \bar{x} \cup \bar{y}$. Correctness follows from construction. Since the complement problem is Π_2^p-complete, C-sufficient feasibility is Σ_2^p-complete.

4 k-Robustness

The notions of robustness described above are based on identifying some "critical" coalition, whose compliance is either necessary and/or sufficient for the correct functioning of the overall normative system. In this section, we explore a slightly different notion, whereby we instead *quantify* the extent to which a normative system is resistant to non-compliance. We introduce the notion of k-*robustness*, where $k \in \mathbb{N}$: intuitively, saying that a normative system is k-robust will mean that it remains effective as long as k *arbitrary* agents comply.

As with C-compliance, we can consider k-compliance from the point of view of both sufficiency and necessity. Where $k \geq 1$, we say a normative system η is k-*sufficient* (w.r.t. some K, φ) if the compliance of *any arbitrary* k *agents* is sufficient to ensure that the normative system is effective with respect to φ. Formally, this involves checking that:

$$\forall C \subseteq A : (|C| \geq k) \qquad \Rightarrow \qquad (K \dagger (\eta \upharpoonright C)) \models \varphi.$$

As with checking C-sufficiency, checking k-sufficiency is hard.

Theorem 5. *Deciding k-sufficiency is co-NP-complete.*

Proof. Membership of co-NP is obvious from the problem definition; for hardness, we reduce TAUT, constructing the Kripke structure, normative system, and objective as in the proof of Theorem 2; and finally, we set $k = 0$. The correctness argument is then as in Theorem 2.

We define the *resilience* of a normative system η (w.r.t. K, φ) as the largest number of non-compliant agents the system can tolerate. Formally, the resilience is the largest number k, $k < n$, such that

$$\forall C \subseteq A : (|C| \leq k) \qquad \Rightarrow \qquad (K \dagger (\eta \upharpoonright C)) \models \varphi.$$

where n is the number of agents. It is easy to see that the resilience of η is the largest number k such that η is $(n - k)$-sufficient. Observe that the resilience is *undefined* iff the objective does not hold even if all agents comply to the norm ($K \dagger \eta \not\models \varphi$). It is immediate that computing the resilience of a normative system is co-NP-complete with respect to Turing reductions.

Example 5. We continue Example 3. While both $\{a, b, c\}$ and $\{a, b, d\}$ are sufficient coalitions, η_1 is not 3-sufficient wrt. K_1, φ_1 because not *every* three-agent coalition is sufficient. It is 4-sufficient (the objective is satisfied if the grand coalition complies). Thus, the resilience is equal to 0.

Now consider the situation where a has left the computer facility; b, c, d remains. Let $K_1', \eta_1', \varphi_1'$ be the corresponding variants of K_1, η_1 and φ_1. Now, each of $\{b, c\}$, $\{b, d\}$ and $\{c, d\}$ are sufficient. Thus, η_1' is 2-sufficient wrt. K_1', φ_1', and the resilience is 1.

We then define k-necessity in the obvious way – η is k-necessary (w.r.t. K, φ) iff:

$$\forall C \subseteq A : (K \dagger (\eta \upharpoonright C)) \models \varphi \qquad \Rightarrow \qquad (|C| \geq k).$$

Theorem 6. *Deciding k-necessity is co-NP-complete.*

Proof. Membership of co-NP is again obvious from the problem definition; for hardness, we reduce SAT to the complement problem, proceeding as in Theorem 3; where l is the number of Boolean variables in the SAT instance, we set $k = l + 1$. Correctness of the reduction is then straightforward.

We say that η is *k-robust*, $k \geq 1$, if it is both k-sufficient and k-necessary. In other words, η is k-robust if it is effective exactly in the event of non-compliance of any arbitrary coalition of up to $n - k$ agents: η is k-robust iff

$$\forall C \subseteq A : (|C| \leq n - k) \qquad \Leftrightarrow \qquad (K \dagger (\eta \upharpoonright C)) \models \varphi.$$

where n is the number of agents. From the results above, it is immediate that checking k-robustness is co-NP-complete.

Example 6. We continue Example 5. While $\{a, b\}$ is the largest necessary coalition, η_1 is 3-necessary wrt. K_1, φ_1 because at least three agents must comply (in this case, either $\{a, b, c\}$ or $\{a, b, d\}$). It is not k-robust for any k, because it is 4-sufficient but not 3-sufficient, and 3-necessary but not 4-necessary.

η_1' is both 2-sufficient and 2-necessary wrt. K_1', φ_1'. It is thus 2-robust. Thus, the objective will be maintained if and only if at least 2 agents comply.

Example 7. We continue Example 6. Consider yet another variant: the agents are again all four a, b, c, d, but their needs have changed. Now each agent only needs a PC, i.e., $\mathit{Useful}(a) = \mathit{Useful}(b) = \mathit{Useful}(c) = \mathit{Useful}(d) = \{R_3\}$. Now we have that no singleton coalition is sufficient and every two-agent coalition is sufficient. The system is 2-sufficient, 2-necessary, 2-robust and its resilience is $4 - 2 = 2$.

The following sums up some general properties of the concepts of k-robustness. Here, "k-sufficient" ("k-necessary") means "k-sufficient (k-necessary) in the context of K and φ".

Proposition 2.

1. *Any system is 0-necessary.*
2. *If the system is k-sufficient, then C is sufficient for any C such that $|C| \geq k$.*
3. *If C is necessary, then the system is $|C|$-necessary.*
4. *If the system is k-sufficient for $k < n$, then no non-empty coalition is necessary.*
5. *k-robustness is unique: if the system is k-robust and k'-robust, then $k = k'$.*

Proof.

1.-3. Immediate.

4. Let $k < n$ and assume that the system is k-sufficient and that $C \neq \emptyset$ is necessary. Let C' be a coalition such that $|C'| \geq k$. By k-sufficiency, $K \dagger (\eta \upharpoonright C') \models \varphi$, and by necessity of C, $C \subseteq C'$. Since C' was arbitrary, we have that $C \subseteq \bigcap_{|C'| \geq j} C'$. Assume that $a \in C$. Let $|C_1| = k$. $a \in C_1$. Now let $b \in A \setminus C_1$ (b exists because $k < n = |A|$), and let $C_2 = C_1 \setminus \{a\} \cup \{\, b\}$. $|C_2| = k$, but $a \notin C_2$ which contradicts the assumption that $a \in C$. Thus, C must be empty.
5. If the system is k-robust and k'-robust for $k > k'$ and C' is a coalition of size k', then by k'-sufficiency $(K \dagger (\eta \upharpoonright C)) \models \varphi$ and by k-necessity it follows that $|C| \geq k$ which is not the case.

5 A Logical Characterisation of Robustness

We have thus far seen two different ways in which we might want to consider robustness: try to identify some "lynchpin" coalition, or try to "quantify" the robustness of the normative system in terms of the number of agents whose compliance is required to make the normative system effective. Often, however, robustness properties will not take either of these forms. For example, here is an argument about robustness that one might typically see: "the system will not overheat as long as at least one sensor works and either one of the relief valves is working or the automatic shutdown is working". Clearly, such an argument does not fit any of the types of robustness property that we have seen so far. So, how are we to characterise such properties? The idea we adopt is to characterise the robustness by means of a *coalition predicate*. Coalition predicates were originally introduced in [3] as a way of quantifying over coalitions. A coalition predicate, as the name suggests, is simply a predicate over coalitions: if P is a coalition predicate, then it denotes a set of coalitions – those that satisfy P.

$eq(C)$	$\hat{=}$	$subseteq(C) \wedge supseteq(C)$
$subset(C)$	$\hat{=}$	$subseteq(C) \wedge \neg eq(C)$
$supset(C)$	$\hat{=}$	$supseteq(C) \wedge \neg eq(C)$
$incl(i)$	$\hat{=}$	$supseteq(\{i\})$
$excl(i)$	$\hat{=}$	$\neg incl(i)$
any	$\hat{=}$	$supseteq(\emptyset)$
$nei(C)$	$\hat{=}$	$\bigvee_{i \in C} incl(i)$
$ei(C)$	$\hat{=}$	$\neg nei(C)$
$gt(n)$	$\hat{=}$	$geq(n+1)$
$lt(n)$	$\hat{=}$	$\neg geq(n)$
$leq(n)$	$\hat{=}$	$lt(n+1)$
$maj(n)$	$\hat{=}$	$geq(\lceil (n+1)/2 \rceil)$
$ceq(n)$	$\hat{=}$	$(geq(n) \wedge leq(n))$

Table 1. Derived coalition predicates.

We first introduce the language of coalition predicates (from [3]), and then show how this language can be used to characterise robustness properties. Syntactically, the language of coalition predicates is built from three atomic predicates $subseteq$, $supseteq$, and geq, and we derive a stock of other predicate forms from these[4]. Formally, the syntax of coalition predicates is given by the following grammar:

$$P ::= subseteq(C) \mid supseteq(C) \mid geq(n) \mid \neg P \mid P \vee P$$

where $C \subseteq A$ is a set of agents and $n \in \mathbb{N}$ is a natural number.

The circumstances under which a coalition $C_0 \subseteq A$ satisfies a coalition predicate P are specified by the satisfaction relation "$\models_{cp}$", defined by the following rules:

$C_0 \models_{cp} subseteq(C)$ iff $C_0 \subseteq C$
$C_0 \models_{cp} supseteq(C)$ iff $C_0 \supseteq C$
$C_0 \models_{cp} geq(n)$ iff $|C_0| \geq n$
$C_0 \models_{cp} \neg P$ iff not $C_0 \models_{cp} P$
$C_0 \models_{cp} P_1 \vee P_2$ iff $C_0 \models_{cp} P_1$ or $C_0 \models_{cp} P_2$

We assume the conventional definitions of implication ($\rightarrow$), biconditional ($\leftrightarrow$), and conjunction ($\wedge$) in terms of $\neg$ and $\vee$. We also find it convenient to make use of the derived predicates defined in Table 1.

Now, given a Kripke structure K, normative system η, objective φ, and coalition predicate P, we say that P *characterises the robustness of* η iff the compliance of any coalition satisfying P is sufficient to ensure that η is effective (w.r.t. K, φ). More formally, P characterises the robustness of η w.r.t. K and φ iff:

$$\forall C \subseteq A : \qquad (C \models_{cp} P) \qquad \Leftrightarrow \qquad ((K \dagger (\eta \upharpoonright C)) \models \varphi).$$

Now, consider the following simple coalition predicate.

$$supseteq(C) \tag{3}$$

[4] In fact, we could choose a smaller base of predicates to work with, deriving the remaining predicates from these, but the definitions would not be succinct; see the discussion in [3].

Expanding out the semantics, we have that (3) characterises the robustness of a normative system η w.r.t. K, φ iff:

$$\forall C' \subseteq A : \qquad (C \subseteq C') \qquad \Leftrightarrow \qquad ((K \dagger (\eta \upharpoonright C)) \models \varphi).$$

In other words, (3) expresses that C are necessary and sufficient. As another simple example, the predicate $geq(k)$ characterises the robustness of η iff η is k-robust. The decision problem of *P-characterisation* is that of checking whether a given coalition predicate P characterises robustness in the way described above. Since we can use P-characterisation to express necessary and sufficient coalitions, we have the following.

Corollary 2. *Deciding P-characterisation is co-*NP*-complete.*

Notice that P-characterisation is fully expressive with respect to robustness properties, in that *any* robustness property can be characterised with a coalition predicate of the form:

$$eq(C_1) \vee eq(C_2) \vee \cdots \vee \; eq(C_u).$$

for some $u \in \mathbb{N}$. In the worst case, of course, we may need a coalition predicate where u may be exponential in the number of agents.

Let us consider some example coalition predicates, and what they say about robustness. Recall the informal example we used in the introduction to this section. Let S be a set of sensors, let R be the set of relief valves, and let a be the automatic shutdown system. Then the following coalition predicate expresses the robustness property expressed in this argument.

$$nei(S) \wedge (nei(R) \vee incl(a))$$

The coalition predicate any expresses the fact that the normative system is trivial, in the sense that it is robust against any deviation (in which case it is unnecessary, since the objective will hold of the original system). The coalition predicate $\neg any$ expresses the fact that the normative system will fail w.r.t. its objective irrespective of who complies with it.

6 Conclusions

We have investigated three types of robustness: necessary and/or sufficient coalitions; the number of non-compliant agents that can be tolerated; and, more generally, a logical characterisation of robustness.

Fitoussi and Tennenholz [6] formulate two criteria when choosing between different social laws. *Simplicity* tries to minimise, for each agent, the differences between states in terms of the allowed actions. The idea behind *minimality* is to reduce the number of forbidden actions that are not necessary to achieve the objective. Obviously, these two criteria typically conflict: one may sacrifice one in favour of the other. One would expect that there is a trade-off between minimality and robustness, and that minimality of η would coincide with the grand coalition A being necessary for it. This match is not perfect, however: first of all, if the latter condition holds, there still may be more transitions forbidden for A than necessary to guarantee the objective φ. Secondly, it might be that not all agents in A are constrained by η. But what we *do* have is that a minimal norm η must have $A(\eta)$ (the agents involved in it) as a necessary coalition.

Recently, French *et al.* proposed a temporal logic of robustness [7]. A brief description of the main ideas, using our formalisms, is as follows. Let η be a norm. A path π complies with η if for no $n \in \mathbb{N}$, $(\pi[n], \pi[n+1]) \in \eta$, i.e., no step in π is forbidden. Let $O\varphi$ mean that φ is obligatory: it is true in s if for all η-compliant s-paths, φ holds. $P\varphi$ (φ is permitted) is $\neg O \neg \varphi$. Given an s-path π, let

$$\begin{aligned} \Delta^1_s(\pi) = \{\pi' \mid \pi' \text{ is } s\text{-path }, \exists j \in \mathbb{N} \forall i < j \pi(i) = \pi'(i) \ \& \\ \pi'[j+1]\pi'[j+2] \ldots \text{ complies with } \eta\} \end{aligned}$$

In words: $\pi' \in \Delta^1_s$ if it is like π up to some point j, in j it may do an illegal step, but from then on complies with the norm. French *et al.* then define an operator $\blacktriangle\varphi$ ('robustly, φ') which is true on a path π, if for all paths in $\Delta^1_s(\pi)$, and π itself, φ is true. So, $\blacktriangle\varphi$ is true in a η-complient path, if it is true in all paths that have at most one η-forbidden transition. This is a way of bringing robustness in to the object language. However, note that in [7], there is no notion of *agency*: only the system can deviate from or comply with a norm. If φ is a universal formula, then $K, s_0 \models P\blacktriangle\varphi$ would imply (in our framework) that there is a single agent i such that $A \setminus \{i\}$ is sufficient for $\mathsf{E}\varphi$, given K and η. Although it seems a good idea for future work to incorporate such 'deontic-like' operators in the object language, even the semantics of [7] is quite different from ours: whereas [7] focusses on the number of illegal transitions, we are concerned with the number of compliant agents, or compliant coalitions.

References

1. T. Ågotnes, W. van der Hoek, J. A. Rodriguez-Aguilar, C. Sierra, and M. Wooldridge. On the logic of normative systems. In *Proc. of the Twentieth Inter. Joint Conf. on Artificial Intelligence (IJCAI-07)*, Hyderabad, India, 2007.
2. T. Ågotnes, W. van der Hoek, and M. Wooldridge. Normative system games. In *Proc. of the Sixth Intern. Joint Conference on Autonomous Agents and Multiagent Systems (AAMAS-2007)*, Honolulu, Hawaii, 2007.
3. T. Ågotnes, W. van der Hoek, and M. Wooldridge. Quantified coalition logic. In *Proc. of the Twentieth Intern. Joint Conf. on Artificial Intelligence (IJCAI-07)*, Hyderabad, India, 2007.
4. R. Axelrod. An evolutionary approach to norms. *American Political Science Review*, 80(4):1095–1110, 1986.
5. E. A. Emerson. Temporal and modal logic. In J. van Leeuwen, editor, *Handbook of Theoretical Computer Science Volume B: Formal Models and Semantics*, pages 996–1072. Elsevier Science Publishers B.V.: Amsterdam, 1990.
6. D. Fitoussi and M. Tennenholtz. Choosing social laws for multi-agent systems: Minimality and simplicity. *Artificial Intelligence*, 119(1-2):61–101, 2000.
7. T. French, C. McCabe-Dansted, and M. Reynolds. A temporal logic of robustness. In B. Konev and F. Wolter, editors, *Frontiers of Combining Systems*, volume 4720 of *LNCS*, pages 193–205, 2007.
8. W. van der Hoek, M. Roberts, and M. Wooldridge. Social laws in alternating time: Effectiveness, feasibility, and synthesis. *Synthese*, 156(1):1–19, May 2007.
9. D. S. Johnson. A catalog of complexity classes. In J. van Leeuwen, editor, *Handbook of Theoretical Computer Science Volume A: Algorithms and Complexity*, pages 67–161. Elsevier Science Publishers B.V.: Amsterdam, 1990.

10. M. J. Osborne and A. Rubinstein. *A Course in Game Theory*. The MIT Press: Cambridge, MA, 1994.
11. Y. Shoham and M. Tennenholtz. On the synthesis of useful social laws for artificial agent societies. In *Proceedings of the Tenth National Conference on Artificial Intelligence (AAAI-92)*, San Diego, CA, 1992.
12. Y. Shoham and M. Tennenholtz. On social laws for artificial agent societies: Off-line design. In P. E. Agre and S. J. Rosenschein, editors, *Computational Theories of Interaction and Agency*, pages 597–618. The MIT Press, 1996.