ISSN 1862 – 4405

# Dagstuhl Seminar Proceedings 07122

# Normative Multi-agent Systems

**Vol. I**

**Guido Boella, Leendert van der Torre, Harko Verhagen (Eds.)**

Dagstuhl Seminar Proceedings 07122

# Normative Multi-agent Systems

G. Boella, L. v. d. Torre, H. Verhagen (Eds.)

Dagstuhl Seminar 07122, 18.03. - 23.03.2007

Vol. I

**ISSN 1862 - 4405**

**07122 Abstracts Collection -- Normative Multi-agent Systems**
**Authors:** Boella, Guido ; Verhagen, Harko ; van der Torre, Leendert

**Introduction to Normative Multiagent Systems**
**Authors:** Boella, Guido ; van der Torre, Leendert ; Verhagen, Harko

**A Game-Theoretic Approach to Normative Multi-Agent Systems**
**Authors:** Boella, Guido ; van der Torre, Leendert

**A Normative Framework for Agent-Based Systems**
**Authors:** Lopez y Lopez, Fabiola ; Luck, Michael ; d'Inverno, Mark

**A Normative Multi-Agent Systems Approach to the Use of Conviviality for Digital Cities**
**Authors:** Caire, Patrice

**Agents, Norms and Forest Cleaning**
**Authors:** Odelstad, Jan

**Aligning Models of Normative Systems and Artificial Societies: Towards norm-governed behavior in virtual enterprises**
**Authors:** Davidson, Paul ; Jacobsson, Andreas

**BIO Logical Agents: Norms, Beliefs, Intentions in Defeasible Logic**
**Authors:** Governatori, Guido ; Rotolo, Antonino

**Choosing Your Beliefs**
**Authors:** Boella, Guido ; da Costa Pereira, Célia ; Pigozzi, Gabriella ; Tettamanzi, Andrea ; van der Torre, Leendert

**Control Patterns in a Health Care Network**
**Authors:** Kartseva, Vera ; Hulstijn, Joris ; Gordijn, Jaap ; Tan, Yao-Hua

**Deriving individual obligations from collective obligations**
**Authors:** Garion, Christophe ; Cholvy, Laurence

**Designing Organizations: Towards a Model**
**Authors:** Bottazzi, Emanuele ; Ferrario, Roberta ; Masolo, Claudio ; Trypuz, Robert

**Emergence In the Loop: Simulating the two way dynamics of norm innovation**
**Authors:** Andrighetto, Giulia ; Conte, Rosaria ; Turrini, Paolo ; Paolucci, Mario

**Epistemic Norms in a Nutshell**
**Authors:** Weydert, Emil

**Expressing and Verifying Business Contracts with Abductive**
**Authors:** Alberti, Marco ; Chesani, Federico ; Gavanelli, Marco ; Lamma, Evelina ; Mello, Paola ; Montali, Marco ; Torroni, Paolo

**Implementing Norms that Govern Non-Dialogical Actions**
**Authors:** Torres da Silva, Viviane

# 07122 Abstracts Collection
# Normative Multi-agent Systems
## — Dagstuhl Seminar —

Guido Boella[1], Harko Verhagen[2] and Leendert van der Torre[3]

[1] Univ. of Torino, IT
guido@di.unito.it
[2] Stockholm Univ., SE
verhagen@dsv.su.se
[3] Univ. of Luxembourg, LU
viviane@fdi.ucm.es

**Abstract.** From 18.03.07 to 23.03.07, the Dagstuhl Seminar 07122 "Normative Multi-agent Systems" was held in the International Conference and Research Center (IBFI), Schloss Dagstuhl. 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.** Normative systems, multi-agent systems

## Introduction to Normative Multiagent Systems

*Harko Verhagen (Stockholm University, S)*

This article introduces the research issues related to and definition of normative multiagent systems.

*Keywords:* Norms, Multiagent systems, Normative multiagent systems

*Joint work of:* Boella, Guido; van der Torre, Leendert; Verhagen, Harko

*Full Paper:* http://drops.dagstuhl.de/opus/volltexte/2007/918

*See also:* G. Boella, L. van der Torre, and H. Verhagen. Introduction to Normative Multiagent Systems, Journal of Computational and Mathematical Organization Theory, 12 (2/3), p. 71 - 80, October 2006.

## Norms of Conversation in a Framework for Agent Communication Languages

*Rodrigo Agerri (University of Birmingham, GB)*

In open and heterogeneous environments offered by the Internet, where agents are designed by different vendors, the development of standards for agent communication needs to keep abreast of new dynamic interaction modalities. The objective of this paper is to contribute to FIPA's standardization effort by proposing a pragmatic approach to the design of agent communication languages (ACLs) in which the meaning of messages is the combination of its semantics and pragmatics. First, we present a reformulation of FIPA's communicative acts (ACL semantics) using a grounded specification language which overcomes some of the usual problems attributed to FIPA's ACL semantics. Then the ACL pragmatics aims to account for the contextual factors that enriches the semantics, such agents' roles, turn-taking, and the satisfiability of messages' perlocutionary effects. We claim that the ACL pragmatics is best specified by means of norms related to agents' obligations, permissions and rights.

*Keywords:* Agent Communication Languages, Norms, Multi-Agent Systems

*Full Paper:* http://drops.dagstuhl.de/opus/volltexte/2007/920

## On the Logic of Normative Systems

*Thomas Agotnes (Bergen University College, N)*

We introduce *Normative Temporal Logic* (NTL), a logic for reasoning about normative systems. NTL is a generalisation of the well-known branching-time temporal logic CTL, in which the path quantifiers $A$ ("on all paths...") and $E$ ("on some path...") are replaced by the indexed deontic operators $O_\eta$ and $P_\eta$, where for example $O_\eta\varphi$ means "$\varphi$ is obligatory in the context of normative system $\eta$".

After defining the logic, we give a sound and complete axiomatisation, and discuss the logic's relationship to standard deontic logics. We present a symbolic representation language for models and normative systems, and identify four different model checking problems, corresponding to whether or not a model is represented symbolically or explicitly, and whether or not we are given an interpretation for the normative systems named in formulae to be checked. We show that the complexity of model checking varies from P-complete up to EXPTIME-hard for these variations.

*Keywords:* Normative systems, normative temporal logic, deontic logic

*Joint work of:* Ågotnes, Thomas; van der Hoek, Wiebe; Rodriguez-Aguilar, Juan A.; Sierra, Carles; Wooldridge, Michael

*Full Paper:* http://drops.dagstuhl.de/opus/volltexte/2007/921

## Expressing and Verifying Business Contracts with Abductive

*Marco Alberti (Università di Ferrara, I)*

In this article, we propose to adopt the SCIFF abductive logic language to specify business contracts, and show how its proof procedures are useful to verify contract execution and fulfilment.

SCIFF is a declarative language based on abductive logic programming, which accommodates forward rules, predicate definitions, and constraints over finite domain variables. Its declarative semantics is abductive, and can be related to that of deontic operators; its operational specification is the sound and complete SCIFF proof procedure, defined as a set of transition rules, which has been implemented and integrated into a reasoning and verification tool. A variation of the SCIFF proof-procedure (g-SCIFF) can be used for static verification of contract properties.

We demonstrate the use of the SCIFF language for business contract specification and verification, in a concrete scenario. In order to accommodate integration of SCIFF with architectures for business contract, we also propose an encoding of SCIFF contract rules in RuleML.

*Keywords:* Contracts, Verification, Abduction

*Joint work of:* Alberti, Marco; Chesani, Federico; Gavanelli, Marco; Lamma, Evelina; Mello, Paola; Montali, Marco; Torroni, Paolo

*Full Paper:* http://drops.dagstuhl.de/opus/volltexte/2007/901

## A Game-Theoretic Approach to Normative Multi-Agent Systems

*Guido Boella (University of Torino, I)*

We explain the raison d'etre and basic ideas of our game-theoretic approach to normative multiagent systems, sketching the central elements with pointers to other publications for detailed developments.

*Keywords:* Normative multiagent systems, deontic logic, input/output logic

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

*Full Paper:* http://drops.dagstuhl.de/opus/volltexte/2007/937

## Normative Multi-Agent Organizations: Modeling, Support and Control, Draft Version

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

In the last years, social and organizational aspects of agency have become a major issue in multi-agent systems' research.

Recent applications of MAS enforce the need of using these aspects in order to ensure some social order within these systems. Tools to control and regulate the overall functioning of the system are needed in order to enforce global laws on the autonomous agents operating in it. This paper presents a normative organization system composed of a normative organization modeling language MOISEInst used to define the normative organization of a MAS, accompanied with SYNAI, a normative organization implementation architecture which is itself regulated with an explicit normative organization specification.

*Keywords:* Organization, multi-agent

*Joint work of:* Boissier, Olivier; Gâteau, Benjamin

*Full Paper:* http://drops.dagstuhl.de/opus/volltexte/2007/902

## Spatially Distributed Normative Objects

*Rafael Bordini (University of Durham, GB)*

Organisational structures for multi-agent systems are usually defined independently of any spatial or temporal structure. Therefore, when the multiagent system is situated in a spatial environment, there is usually a conceptual gap between the definition of the system's organisational structures and the definition of the environment. In this paper, we focus on a mechanism for the spatial distribution of an organization's normative information. Spatially distributing the normative information over the environment is a natural way to simplify the definition of organisational structures and the development of large-scale multi-agent systems. By distributing the normative information in different spatial locations, we allow agents to directly access the relevant information needed in each environmental context. We extend our previous work on a language for modelling multi-agent environments in order to allow for the definition of spatially distributed norms in the form of normative objects.

*Keywords:* Multi-Agent Systems, Environment Modelling, Normative Infrastructure

*Joint work of:* Okuyama, Fabio; Bordini, Rafael; Rocha Costa, Antônio Carlos

*Full Paper:* http://drops.dagstuhl.de/opus/volltexte/2007/903

*See also:* F. Y. Okuyama, R. H. Bordini, and A. C. da Rocha Costa. Spatially Distributed Normative Objects. In G. Boella et al. (edts.), Proceedings of the International Workshop on Coordination, Organization, Institutions and Norms in Agent Systems (COIN), held with ECAI, Riva del Garda, Italy, 28 August 2006.

## Designing Organizations: Towards a Model

*Emanuele Bottazzi (Institute of Cognitive Sciences & Technology-Trent, I)*

The purpose of this paper is to draw a preliminary model of an ontology of organizations. The emphasis is on the structural aspects of organizations and the relations that these have with the design process of the organization itself on the one hand, and with its normative layer on the other.

*Keywords:* Ontology, organizations, structure, design, norms

*Joint work of:* Bottazzi, Emanuele; Ferrario, Roberta; Masolo, Claudio; Trypuz, Robert

*Full Paper:* http://drops.dagstuhl.de/opus/volltexte/2007/904

## What an Agent Ought To Do

*Jan M. Broersen (Utrecht University, NL)*

This paper reviewes Horty's 2001 book 'Agency and Deontic Logic'. We place Horty's research I a broader context and discuss the relevancy for logics for multi-agent systems.

*Keywords:* Deontic logic, STIT, agency, action

*Joint work of:* Broersen, Jan M.; van der Torre, Leendert

*Full Paper:* http://drops.dagstuhl.de/opus/volltexte/2007/905

*Full Paper:*
http://ipsapp007.kluweronline.com/ips/frames/frames.asp?J=4521&cookie=1

## A Normative Multi-Agent Systems Approach to the Use of Conviviality for Digital Cities

*Patrice Caire (University of Luxemburg, L)*

Conviviality is a mechanism to reinforce social cohesion and a tool to reduce miscoordination between individuals, groups and institutions in web communities, for example in digital cities. We use a two-fold definition of conviviality as a condition for social interactions and an instrument for the internal regulation of social systems. In this paper we discuss the use of normative multi-agent systems to analyze the use of conviviality for digital cities, by contrasting norms for conviviality with legal and institutional norms in digital cities. We show the role of the distinction among various kinds of norms, the explicit representation of norms, the violability of norms, the dynamics of norms and the creation of norms in the context of conviviality.

*Keywords:* Conviviality, multi-agent systems, normative systems, social computing, digital cities.

*Full Paper:* http://drops.dagstuhl.de/opus/volltexte/2007/906

## Aligning Models of Normative Systems and Artificial Societies: Towards norm-governed behavior in virtual enterprises

*Paul Davidson (Blekinge Institute of Technology - Karlskrona, S)*

The purpose is to explore how norm-governed behavior within agent societies can be achieved in the context of Virtual Enterprises. We analyze a number of formal models from the agent research field, of which three models focus on the society aspects and three models focus on norms. A general observation is that the models reviewed are not concordant with each other and therefore require further alignment. A number of additions that may enrich the norm-focused models are suggested. It is also concluded that the introduction of different types of norms on different levels can be applied to ensure sound collaboration in agent-supported virtual enterprises. Moreover, the deployment of norm defender and promoter functionality is suggested to ensure norm compliance and punishments of norm violations.

*Keywords:* Agents, norms, virtual enterprises

*Joint work of:* Davidson, Paul; Jacobsson, Andreas

*Full Paper:* http://drops.dagstuhl.de/opus/volltexte/2007/908

## Specifying and Enforcing Norms in Artificial Institutions

*Nicoletta Fornara (University of Lugano, CH)*

In this paper we investigate two important and related aspects of the formalization of open interaction systems: how to specify norms, and how to enforce them by means of sanctions. The problem of specifying the sanctions associated with the violation of norms is crucial in an open system because, given that the compliance of autonomous agents to obligations and prohibitions cannot be taken for granted, norm enforcement is necessary to constrain the possible evolutions of the system, thus obtaining a degree of predictability that makes it rational for agents to interact with the system. In our model, norms are specified declaratively. When certain events take place, norms become active and generate pending commitments for the agents playing certain roles. Norms also specify the sanctions associated with their violation. In the paper, we analyze the concept of sanction in detail and propose a mechanism through which sanctions can be applied.

*Keywords:* Norms, Sanctions, Commitments, Artificial Institutions

*Joint work of:* Fornara, Nicoletta; Colombetti, Marco

*Full Paper:* http://drops.dagstuhl.de/opus/volltexte/2007/909

## Norms and plans as unification criteria for social collectives

*Aldo Gangemi (Institute of Cognitive Sciences & Technology - Rom, I)*

Based on the formal-ontological paradigm of Constructive Descriptions and Situations, we propose a definition of social collectives that includes social agents, plans, norms, and the conceptual relations between them. We also propose a typology of social collectives, including collection of agents, knowledge community, intentional collective, and intentional normative collective. Our ontology, represented as a first-order theory, provides the expressivity to talk about the contexts (social, informational, circumstantial, and epistemic), in which collectives make and produce sense

*Keywords:* Formal Ontology, Constructivism, Social Entities, Semantic Web

*Joint work of:* Gangemi, Aldo; Lehmann, Jos; Catenacci, Carola

*Full Paper:* http://drops.dagstuhl.de/opus/volltexte/2007/910

## Deriving individual obligations from collective obligations

*Christophe Garion (SUPAERO-Toulouse, F)*

A collective obligation is an obligation directed to a group of agents so that the group, as a whole, is obliged to achieve a given task.

The problem investigated here is the impact of collective obligations to individual obligations, i.e. obligations directed to single agents of the group. The groups we consider do not have any particular hierarchical structure nor have an institutionalized representative agent. In this case, we claim that the derivation of individual obligations from collective obligations depends on several parameters among which the ability of the agents (i.e. what they can do) and their own personal commitments (i.e. what they are determined to do). As for checking if these obligations are fulfilled or not, we need to know what are the actual actions performed by the agents.

This present paper addresses these questions in the rather general case when the collective obligations are conditional ones.

*Keywords:* Deontic logic, action, representation of preferences

*Joint work of:* Garion, Christophe; Cholvy, Laurence

*Full Paper:* http://drops.dagstuhl.de/opus/volltexte/2007/911

*See also:* Proceedings of the Second International Joint Conference on Autonomous Agents and Multiagent Systems, pages 962-964 (Poster). ACM Press, 2003

## Towards a General Framework for Modelling Roles

*Valerio Genovese (University of Torino, I)*

Role is a widespread concept, it is used in many areas like MAS, Programming Languages, Organizations, Security and OO mod- elling. Unfortunately, it seems that the literature is not actually able to give a uniform definition of roles, there exist several approaches that model roles in many different (or even opposite) ways. In this draft we start to define a meta-model for roles. Our aim is to build a formal framework through which we can describe different roles appeared in the literature or implemented in up and running computer systems. In particular we give a new definition of role's foundation introducing ses- sions, which are a formal instrument to talk about role's states and we show how sessions may be useful to model many different role's accounts.

*Keywords:* Roles, Organizations, Object OrientedModelling, Multi-Agent Systems, Security

*Full Paper:* http://drops.dagstuhl.de/opus/volltexte/2007/925

## BIO Logical Agents: Norms, Beliefs, Intentions in Defeasible Logic

*Guido Governatori (The University of Queensland, AU)*

In this paper we follow the BOID (Belief, Obligation, Intention, Desire) architecture to describe agents and agent types in Defeasible Logic. We argue, in particular, that the introduction of obligations can provide a new reading of the concepts of intention and intentionality. Then we examine the notion of social agent (i.e., an agent where obligations prevail over intentions) and discuss some computational and philosophical issues related to it.

We show that the notion of social agent either requires more complex computations or has some philosophical drawbacks.

*Keywords:* Social Agents, Defeasible Logic, Complexity of Agents

*Joint work of:* Governatori, Guido; Rotolo, Antonino

*Full Paper:* http://drops.dagstuhl.de/opus/volltexte/2007/912

## On the logic of constitutive rules

*Davide Grossi (Utrecht University, NL)*

The paper proposes a logical systematization of the notion of counts-as which is grounded on a very simple intuition about what counts-as statements actually mean, i.e., forms of classification. Moving from this analytical thesis the paper disentangles three semantically different readings of statements of the type X counts as Y in context C, from the weaker notion of contextual classification to the stronger notion of constitutive rule. These many ways in which counts-as can be said are then formally addressed by making use of modal logic techniques. The resulting framework allows for a formal characterization of all the involved notions and their reciprocal logical relationships.

*Keywords:* Constitutive rules, counts-as, modal logic

*Joint work of:* Grossi, Davide; Meyer, John-Jules; Dignum, Frank

*Full Paper:* http://drops.dagstuhl.de/opus/volltexte/2007/913

## Prioritized Conditional Imperatives: Problems and a New Proposal

*Jörg Hansen (Universität Leipzig - ICCAS, D)*

The sentences of deontic logic may be understood as describing what an agent ought to do when faced with a given set of norms. If these norms come into conflict, the best the agent can be expected to do is to follow a maximal subset of the norms. Intuitively, a priority ordering of the norms can be helpful in determining the relevant sets and resolve conflicts, but a formal resolution mechanism has been difficult to provide. In particular, reasoning about prioritized conditional imperatives is overshadowed by problems such as the 'order puzzle' that are not satisfactorily resolved by existing approaches. The paper provides a new proposal as to how these problems may be overcome.

*Keywords:* Deontic logic, default logic, priorities, logic of imperatives

*Full Paper:* http://drops.dagstuhl.de/opus/volltexte/2007/914

## Control Patterns in a Health Care Network

*Joris Hulstijn (Vrije Universiteit Amsterdam, NL)*

In this paper we present control patterns for the analysis and design of administrative control mechanisms in a network organization.

A control pattern is a description of a generic and reusable control mechanism that solves a specific control problem, to be selected on the basis of the context. To represent the context and solution, we analyze a network organization as a set of actors who transfer objects of economic value. The usefulness and adequacy of the control patterns is demonstrated by a case study of the governance and control mechanisms of the Dutch public health insurance network for exceptional medical expenses (AWBZ).

*Keywords:* Governance and control, network organizations, value modeling

*Joint work of:* Kartseva, Vera; Hulstijn, Joris; Gordijn, Jaap; Tan, Yao-Hua

*Full Paper:* http://drops.dagstuhl.de/opus/volltexte/2007/915

## Norms and accountability in multi-agent societies

*Rodger Kibble (Goldsmiths College - London, GB)*

It is argued that norms are best understood as classes of constraints on practical reasoning, which an agent may consult either to select appropriate goals or commitments according to the circumstances, or to construct a discursive justification for a course of action after the event.

We also discuss the question of how norm-conformance can be enforced in an open agent society, arguing that some form of peer pressure is needed in open agent societies lacking universally-recognised rules or any accepted authority structure. The paper includes formal specifications of some data structures that may be employed in reasoning about normative agents.

*Keywords:* Norms, agents, social commitments, reasoning

*Full Paper:* http://drops.dagstuhl.de/opus/volltexte/2007/916

## A Normative Framework for Agent-Based Systems

*Fabiola Lopez y Lopez (Benemérita Universidad Autónoma de Puebla, MEX)*

One of the key issues in the computational representation of open societies relates to the introduction of norms that help to cope with the heterogeneity, the autonomy and the diversity of interests among their members. Research regarding this issue presents two omissions. One is the lack of a canonical model of norms that facilitates their implementation, and that allows us to describe the processes of reasoning about norms. The other refers to considering, in the model of normative multi-agent systems, the perspective of individual agents and what they might need to effectively reason about the society in which they participate. Both are the concerns of this paper, and the main objective is to present a formal normative framework for agent-based systems that facilitates their implementation.

*Keywords:* Normative agents, normative multi-agent systems

*Joint work of:* Lopez y Lopez, Fabiola; Luck, Michael; d'Inverno, Mark

*Full Paper:* http://drops.dagstuhl.de/opus/volltexte/2007/933

*Full Paper:*
http://dx.doi.org/10.1007/s10588-006-9545-7

*See also:* Computational & Mathematical Organization Theory. 12:227-250. Springer. 2006

## Towards a Logic of Graded Normativity and Norm Adherence

*Matthias Nickles (TU München, D)*

A key focus of contemporary agent-oriented research and engineering is on open multiagent systems composed of truly autonomous, interacting agents. This poses new challenges, as entities in open systems are usually more or less mentally opaque (e.g., possibly insincere), and can enter and leave the system at will. Thus interactions among such black- or gray-box entities usually imply more or less severe contingencies in behavior: Among other issues, in principle, the adherence of agents to norms cannot be guaranteed in such systems. As a response to this issue, this paper proposes a logic-based approach based on the notion of (possibly probabilistic) behavioral expectations, which are stylized either as adaptive (i.e., predictive) or normative (i.e., prescriptive). Some features of this approach are the enabling of "soft norms" which are automatically weakened to some degree if contradicted at runtime, and the possibility to quantify norm adherence using the measurement of norm deviance.

*Keywords:* Computational Norms, Dynamic Logic, Computational Expectations, Social AI, Belief Revision

*Full Paper:* http://drops.dagstuhl.de/opus/volltexte/2007/926

## Agents, Norms and Forest Cleaning

*Jan Odelstad (University of Gävle, S)*

The automation of forest cleaning presupposes principles for choosing those trees that ought to be taken away and those that shall be left standing. In this paper, which is a report on a work in progress, 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 that the agent system can evaluate the efficiency of the normative system and the utility function and, furthermore, suggest improvements of them. Earlier works on norms and norm-regulation of agent systems that the author has been involved in are used to elucidate the problem area discussed in the paper.

*Keywords:* Norm, normative system, norm-regulated agent, forest cleaning

*Full Paper:* http://drops.dagstuhl.de/opus/volltexte/2007/917

## Emergence In the Loop: Simulating the two way dynamics of norm innovation

*Mario Paolucci (ISTC-CNR - Rom, I)*

In this paper we will present the EMIL project, "EMergence In the Loop: Simulating the two-way dynamics of norm innovation", a three-year project funded by the European Commission (Sixth Framework Programme -Information Society and Technologies) in the framework of the initiative "Simulating Emergent Properties in Complex Systems". The EMIL project intends to contribute to the study of social complex systems by modelling norm innovation as a phenomenon implying interrelationships among multiple levels. It shall endeavour to point out that social dynamics in societies of intelligent agents is necessarily bidirectional, which adds complexity to the emergence processes. The micro-macro link will be modelled and observed in the emergence of properties at the macro-level and their immergence into the micro-level units. The main scientific aim of the EMIL project is to construct a simulator for exploring and experimenting norm-innovation.

*Keywords:* Norm innovation, emergence, immergence, simulation, social complexity

*Joint work of:* Paolucci, Mario; Andrighetto, Giulia; Conte, Rosaria; Turrini, Paolo

*Full Paper:* http://drops.dagstuhl.de/opus/volltexte/2007/907

## Ten Philosophical Problems in Deontic Logic

*Gabriella Pigozzi (University of Luxemburg, L)*

The paper discusses ten philosophical problems in deontic logic: how to formally represent norms, when a set of norms may be termed 'coherent', how to deal with normative conflicts, how contrary-to-duty obligations can be appropriately modeled, how dyadic deontic operators may be redefined to relate to sets of norms instead of preference relations between possible worlds, how various concepts of permission can be accommodated, how meaning postulates and counts-as conditionals can be taken into account, and how sets of norms may be revised and merged. The problems are discussed from the viewpoint of input/output logic as developed by van der Torre Makinson. 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.

*Keywords:* Deontic logic, normative systems, input/output logic

*Joint work of:* Hansen, Jörg; Pigozzi, Gabriella; van der Torre, Leendert

*Full Paper:* http://drops.dagstuhl.de/opus/volltexte/2007/941

## Interaction between Normative Systems and Cognitive agents in Temporal Modal Defeasible Logic

*Regis Riveret (Università di Bologna, I)*

While some recent frameworks on cognitive agents addressed the combination of mental attitudes with deontic concepts, they commonly ignore the representation of time. We propose in this paper a variant of Temporal Modal Defeasible Logic to deal in particular with temporal intervals.

*Keywords:* Time, Norm, Temporal Modal Defeasible Logic

*Joint work of:* Riveret, Regis; Rotolo, Antonino; Governatori, Guido

*Full Paper:* http://drops.dagstuhl.de/opus/volltexte/2007/923

## Implementing Norms that Govern Non-Dialogical Actions

*Viviane Torres da Silva (Univ. Comp. de Madrid, E)*

The governance of open multi-agent systems is particular important since those systems are composed by heterogeneous, autonomous and independently designed agents. Such governance is usually provided by the establishment of norms that regulate the actions of agents. Although there are several approaches that formally describe norms, there are still few of them that propose their implementation. In additions, only one that provides support for implementing norms deals with non-dialogical actions since the others only deal with dialogical actions, i.e., actions that provide the interchange of messages between agents. In this paper we propose the implementation of norms that govern non-dialogical actions by extending one of the approaches that regulate dialogical ones. Non-dialogical actions are not related to the interactions between agents but to tasks executed by agents that characterize, for instance, the access to resources, their commitment to play roles or their movement into environments and organizations.

*Keywords:* Norm, governance of multi-agent system, non-dialogical action

*Full Paper:* http://drops.dagstuhl.de/opus/volltexte/2007/927

## Normtypologies

*Harko Verhagen (Stockholm University, S)*

In this extended abstract I describe some norm typolgies developed within sociololgy and social philosophy. Using these typologies we can determine the boundaries of the different approaches to normative agent systems.

*Keywords:* Norms,Multiagent systems, Normative multiagent systems

*Full Paper:* http://drops.dagstuhl.de/opus/volltexte/2007/930

## Epistemic Norms in a Nutshell

*Emil Weydert (University of Luxemburg, L)*

We present some thoughts on epistemic norms.

*Keywords:* Epistemic norms, Trust, Meta-Science

*Extended Abstract:* http://drops.dagstuhl.de/opus/volltexte/2007/924

## Choosing Your Beliefs

*Célia da Costa Pereira (Universitá di Milano, I)*

This paper presents and discusses a novel approach to indeterministic belief revision. An indeterministic belief revision operator assumes that, when an agent is confronted with a new piece of information, it can revise its belief sets in more than one way. We define a rational agent not only in terms of what it believes but also of what it desires and wants to achieve. Hence, we propose that the agent's goals play a role in the choice of (possibly) one of the several available revision options. Properties of the new belief revision mechanism are also investigated.

*Keywords:* Rational agents, indeterministic belief revision, qualitative decision theory

*Joint work of:* Boella, Guido; da Costa Pereira, Célia; Pigozzi, Gabriella; Tettamanzi, Andrea; van der Torre, Leendert

*Full Paper:* http://drops.dagstuhl.de/opus/volltexte/2007/938

## What is Input/Output Logic? Input/Output Logic, Constraints, Permissions

*Leon van der Torre (University of Luxemburg, L)*

We explain the *raison d'être* and basic ideas of input/output logic, sketching the central elements with pointers to other publications for detailed developments. The motivation comes from the logic of norms. Unconstrained input/output operations are straightforward to define, with relatively simple behaviour, but ignore the subtleties of contrary-to-duty norms. To deal with these more sensitively, we constrain input/output operations by means of consistency conditions, expressed via the concept of an outfamily.

They also provide a convenient platform for distinguishing and analysing several different kinds of permission.

*Keywords:* Deontic logic, input/output logic, constraints, permissions

*Joint work of:* Makinson, David; van der Torre, Leendert

*Full Paper:* http://drops.dagstuhl.de/opus/volltexte/2007/928

# Introduction to Normative Multiagent Systems

Guido Boella[1], Leendert van der Torre[2] and Harko Verhagen[3]

[1] Dipartimento di Informatica, Università di Torino
I-10149, Torino, Corso Svizzera 185, Italy
guido@di.unito.it
[2] Department of Computer Science and Communications, University of Luxembourg
L-1359, Luxembourg, rue Richard Coudenhove 6 - Kalergi, Luxembourg
leon.vandertorre@uni.lu
[3] Department of Computer and Systems Sciences, Stockholm University / KTH
SE-16440, Kista, Forum 100, Sweden
verhagen@dsv.su.se

**Abstract.** This article introduces the research issues related to and definition of normative multiagent systems.

**Keywords.** Norms, Multiagent systems, Normative multiagent systems

## 1 Introduction

Normative multiagent systems as a research area can be defined as the intersection of normative systems and multiagent systems. 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.

With 'normative' we mean 'conforming to or based on norms', as in *normative behavior* or *normative judgments*. According to the Merriam-Webster Online [1] 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.

Normative multiagent systems are an example of the use of sociological theories in multiagent systems, and more generally of the relation between agent theory and the social sciences such as sociology, philosophy, economics, and legal

Dagstuhl Seminar Proceedings 07122
Normative Multi-agent Systems
http://drops.dagstuhl.de/opus/volltexte/2007/918

science. The need for social science theories and concepts like norms in multiagent systems is now well established. For example, Wooldridge's weak notion of agency is based on flexible autonomous action [2], and social ability as the interaction with other agents and co-operation is one of the three meanings of flexibility; the other two are reactivity as interaction with the environment, and pro-activeness as taking the initiative. In this definition autonomy refers to non-social aspects, such as operating without the direct intervention of humans or others, and have some kind of control over their actions and internal state. For some other arguments for the need for social theory in multiagent systems, see, for example, [3,4,5]. For a more complete discussion on the need of social theory in general, and norms in particular, see the AgentLink roadmap [6].

Social concepts like norms are important for multiagent systems, because multiagent system research and sociology share the interest in the relation between micro-level agent behaviour and macro-level system effects. In sociology this is the (in)famous micro-macro link [7] that focuses on the relation between individual agent behaviour and characteristics at the level of the social system. In multiagent system research, this boils down to the question "How to ensure efficiency at the level of the multiagent system whilst respecting individual autonomy?". According to Verhagen [8] three possible solutions to this problem comprise of the use of central control which gravely jeopardizes the agent's autonomy, internalized control like the use of social laws [9], and structural coordination [10] including learning norms.

Before we discuss normative multiagent systems, we consider some discussions on norms in the social sciences.

## 2 Norms and normative systems

In the 1960's, the sociologist Gibbs [11] wrote an influential article on the problems concerning the definition and classification of norms, and observes that the various types of norms involve "a collective evaluation of behavior in terms of what it *ought* to be; a collective expectation as to what behavior *will be*; and/or particular *reactions* to behavior, including attempts to apply sanctions or otherwise induce a particular kind of conduct." [11, p. 589, original emphasis]

More recently, Therborn [12] presented an overview of the role of norms for social theory and analysis. Normative action is based upon wanting to do the right thing rather than the thing that leads to ends or goals, which he calls teleological action, or the thing that leads to, expresses, or is caused by an emotion, called emotional action.

Therborn distinguishes among 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 equaled with role plays, i.e., roles find their expressions in expectations, obligations, and rights vis-a-vis the role holder's behaviour.

Therborn also addresses the dynamics and changing of norms. The dynamics of norms at the level of the individual agent is how norms are learned or propagated in a population. Socialization is based on identification, perceiving the compliance with the norms by other agents, or the entering of an institution. Norms are (re)enforced by the presence of incentives or sanctions. Changes in either of these three three socialization mechanisms lead to changes in the set of norms of the individual agent. These changes may be inhibited either by changes in the social system or changed circumstances, or by changes in the interpretation of the norms by the agents within the system.

Within philosophy normative systems have traditionally been studied by moral and legal philosophers. Alchourròn and Bulygin [13] argue 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 $\alpha$ there is at least one normative correlation, we shall say that the set $\alpha$ has normative consequences. A system of sentences which has some normative consequences will be called a normative system." [13, p.55].

In computer science, Meyer and Wieringa define normative systems as "systems in the behavior of which norms play a role and which need normative concepts in order to be described or specified" [14, preface]. They also explain why normative systems are intimately related with deontic logic.

> "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! Deontic logic provides a means to do just this by using special modal operators that indicate the status of behavior: that is whether it is legal (normative) or not" [14, preface].

## 3 Normative multiagent systems

The agents in the environment of a normative system interact with the normative system in various ways. First, from the perspective of the agents, agents can create new norms, update or maintain norms, and enforce norms, using roles defined in the normative system such as legislators or policemen. Secondly, from the perspective of social order, we can also look at the interaction between the normative system and its environment from the viewpoint of the normative

system. In this viewpoint, the normative system uses the agents playing a role in it – the legislators, policemen and the like – to maintain an equilibrium in the normative multiagent system. In this perspective, we can distinguish at least two levels of equilibrium. First, norms are used to maintain social order in a normative multiagent system. Second, normative system contain a mechanism for updating themselves, to adapt to changing circumstances in its environment.

Jones and Carmo [15] define a normative system as "Sets of agents whose interactions are 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." In our opinion, this is too general, as a normative system does not contain the agents themselves. It also is not a satisfactory definition of normative multiagent system, because it precludes the agents' control over the set of norms. We therefore use the following definition in this paper.

> 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.

Note that this definition makes no presumptions about the internal workings of an agent nor of the way norms find their expression in agent's behaviour.

Since norms are explicitly represented, according to our definition of a normative multiagent system, the question should be raised how norms are represented. Norms can be interpreted as a special kind of constraint, and represented depending on the domain in which they occur. However, the representation of norms by domain dependent constraints runs into the question what happens when norms are violated. Not all agents behave according to the norm, and the system has to deal with it. In other words, norms are not hard constraints, but soft constraints. For example, the system may sanction violations or reward good behavior. Thus, the normative system has to monitor the behavior of agents and enforce the sanctions. Also, when norms are represented as domain dependent constraints, the question will be raised how to represent permissive norms, and how they relate to obligations. Whereas obligations and prohibitions can be represented as constraints, this does not seem to hold for permissions. For example, how to represent the permission to access a resource under an access control system? Finally, when norms are represented as domain dependent constraints, the question can be raised how norms evolve.

We therefore believe that norms should be represented as a domain independent theory, for example in deontic logic [16,17,18,19,20,21]. Deontic logic studies logical relations among obligations and permissions, and more in particular violations and contrary-to-duty obligations, permissions and their relation to obligations, and the dynamics of obligations over time. Therefore, insights from deontic logic can be used to represent and reason with norms. Deontic logic also offers representations of norms as rules or conditionals. However, there

are several aspects of norms which are not covered by constraints nor by deontic logic, such as the relation between the cognitive abilities of agents and the global properties of norms.

Conte, Falconi and Sartor [22] say that normative multiagent systems research focuses on two different sets of problems. On the one hand, they claim that legal theory and deontic logic supply a theory for of norm-governed interaction of autonomous agents while at the same time lacking a model that integrates the different social and normative concepts of this theory. On the other hand, they claim that three other problems are of interest in multiagents systems research on norms: how agents can acquire norms, how agents can violate norms, and how an agent can be autonomous. For artificial agents, norms can be designed as in legal human systems, forced upon, for example when joining an institution, or they can emerge from the agents making them norm autonomous [8]. Agent decision making in normative systems and the relation between desires and obligations has been studied in agent architectures [23], which thus explain how norms and obligations influence agent behavior.

An important question is where norms come from. Norms are not necessarily created by a single legislator, they can also emerge spontaneously, or be negotiated among the agents. In electronic commerce research, for example, cognitive foundations of social norms and contracts are studied [24]. Protocols and social mechanisms are now being developed to support such creations of norms in multiagent systems. When norms are created, the question how they are enforced can be raised. For example, when a contract is violated, the violator may have to pay a penalty. But then there has to be a monitoring and sanctioning system, for example police agents in an electronic institution. Such protocols or roles in a multiagent system are part of the construction of social reality, and Searle [25] has argued that such social realities are constructed by constitutive norms. This again raises the question how to represent such constitutive or counts-as norms, and how they are related to regulative norms like obligations and permissions [24].

Not only the relation between norms and agents must be studied, but also the relation between norms and other social and legal concepts. How do norms structure organizations? How do norms coordinate groups and societies? How about the contract frames in which contracts live? How about the legal contexts in which contract frames live? How about the relation between legal courts? Though in some normative multiagent systems there is only a single normative system, there can also be several of them, raising the question how normative systems interact. For example, in a virtual community of resource providers each provider may have its own normative system, which raises the question how one system can authorize access in another system, or how global policies can be defined to regulate these local policies [26].

Summarizing, normative multiagent systems study general and domain independent properties of norms. It builds on results obtained in deontic logic, the logic of obligations and permissions, for the representation of norms as rules, the application of such rules, contrary-to-duty reasoning and the relation to per-

missions. However, it goes beyond logical relations among obligations and permissions by explaining the relation among social norms and obligations, relating regulative norms to constitutive norms, explaining the evolution of normative systems, and much more.

Some of these issues can be discussed in more detail. These include action (e.g., the BDI model of agency) with models of normative action, to be combined, reasoning and dynamics, and theories of normative action into implementable formal models.

General themes that are to be addressed in research on normative agent systems include

1. intra-agent aspects of norms,
2. interagent aspects of norms,
3. normative systems and their borders, and
4. combining normative systems.

In [27] a collection of articles on these issues is presented.

## References

1. Merriam-Webster OnLine: The Language Center (2005) www.m-w.com/.
2. Wooldridge, M.: An Introduction to MultiAgent Systems. Wiley (2002)
3. Bond, A.H., Gasser, L.: An Analysis of Problems and Research in DAI. In Bond, A.H., Gasser, L., eds.: Readings in Distributed Artificial Intelligence, Morgan Kaufmann (1988) 3–35
4. Conte, R., Gilbert, N.: Computer Simulation for Social Theory. In: Computer Simulation for Social Theory. UCL Press. (1995) 1 – 18
5. Verhagen, H., Smit, R.: Modelling Social Agents in a Multiagent World. In Van de Velde, W., Perram, J.W., eds.: , Position Papers MAAMAW 1996, Technical Report 96-1, Vrije Universiteit Brussel - Artificial Intelligence Laboratory (1996)
6. : Agent Technology Roadmap: A Roadmap for Agent-Based Computing. (2005)
7. Alexander, J., Giesen, B., Münch, R., Smelser, N., eds.: The Micro-Macro Link. University of California Press (1987)
8. Verhagen, H.: Norm Autonomous Agents. PhD thesis, Department of System and Computer Sciences, The Royal Institute of Technology and Stockholm University, Sweden (2000)
9. Shoham, Y., Tennenholtz, M.: On the Synthesis of Useful Social Laws for Artificial Agent Societies (Preliminary Report). In: Proceedings of the National Conference on Artificial Intelligence, San Jose, CA (1992) 276–281
10. Ossowski, S.: Co-ordination in Artificial Agent Societies. Springer (1999)
11. Gibbs, J.P.: Norms: The Problem of Definition and Classification. The American Journal of Sociology **70** (1965) 586 – 594
12. Therborn, G.: Back to Norms! On the Scope and Dynamics of Norms and Normative Action. Current Sociology **50** (2002) 863 – 880
13. Alchourròn, C., Bulygin, E.: Normative Systems. Springer (1971)
14. Meyer, J.J., Wieringa, R., eds.: Deontic Logic in Computer Science: Normative System Specification. Wiley (1993)
15. Jones, A., Carmo, J.: Deontic Logic and Contrary-to-Duties. In Gabbay, D., ed.: Handbook of Philosophical Logic. Kluwer (2001) 203279

16. von Wright, G.: Mind. **60** (1951) 1–15
17. van der Torre, L., Tan, Y.: Contrary-to-duty reasoning with preference-based dyadic obligations. Annals of Mathematics and Artificial Intelligence **27** (1999) 49–78
18. van der Torre, L.: Contextual deontic logic: Normative agents, violations and independence. Annals of Mathematics and Artificial Intelligence **37(1-2)** (2003) 33–63
19. Makinson, D., van der Torre, L.: Input-output logics. Journal of Philosophical Logic **29** (2000) 383–408
20. Makinson, D., van der Torre, L.: Constraints for input-output logics. Journal of Philosophical Logic **30(2)** (2001) 155–185
21. Makinson, D., van der Torre, L.: Permissions from an input/output perspective. Journal of Philosophical Logic **32(4)** (2003) 391–416
22. Conte, R., Falcone, R., Sartor, G.: Introduction: Agents and Norms: How to Fill the Gap? Artificial Intelligence and Law (1999) 1 – 15
23. Broersen, J., Dastani, M., Hulstijn, J., van der Torre, L.: Goal generation in the BOID architecture. Cognitive Science Quarterly **2(3-4)** (2002) 428–447
24. Boella, G., van der Torre, L.: A game theoretic approach to contracts in multiagent systems. IEEE Trans. SMC, Part C (2006)
25. Searle, J.R.: The Construction of Social Reality. The Free Press (1995)
26. Boella, G., van der Torre, L.: Security policies for sharing knowledge in virtual communities. IEEE Trans. SMC, Part A (2006)
27. Boella, G., Van der Torre, L., Verhagen, H., eds.: Special Issue on Normative Multiagent Systems - Journal of Computational & Mathematical Organization Theory. Volume 12 (2 - 3). (2006)

# A Game-Theoretic Approach to Normative Multi-Agent Systems

Guido Boella[1] and Leendert van der Torre[2]

[1] Università di Torino, Dipartimento di Informatica
10149, Torino, Cso Svizzera 185, Italia
guido@di.unito.it
[2] University of Luxembourg, Computer Science and Communications (CSC)
1359, Luxembourg, 6 rue Richard Coudenhove Kalergi, Luxembourg
leendert@vandertorre.com

**Abstract.** We explain the *raison d'être* and basic ideas of our game-theoretic approach to normative multiagent systems, sketching the central elements with pointers to other publications for detailed developments.

**Keywords.** Normative multiagent systems, deontic logic, input/output logic

## Introduction

We explain the *raison d'être* and basic ideas of our game-theoretic approach to normative multi-agent systems, sketching the central elements with pointers to other publications for detailed developments. In particular, we address the following questions:

**Motivation.** Why do we need a game-theoretic approach to normative multi-agent systems?
**Objectives.** What do we want to achieve with the theory of normative multi-agent systems?
**Methodology.** How do we achieve the objectives?
**Results.** Which results have been obtained thus far?
**Interdisciplinarity.** How are various disciplines used in the theory?

We aim to explain our own approach, and we are therefore very brief with respect to recent related approaches in the area of normative multiagent systems. For these other approaches, see the special issue on normative multiagent systems in *Computational and Mathematical Organization Theory* [68], these DROPS proceedings, the proceedings of the biannual workshops on deontic logic in computer science ($\Delta$EON) and of the COIN workshop series.[1]

The layout of this paper follows the five questions above, addressing each of them in a new section.

[1] http://www.ia.urjc.es/COIN2007/

Dagstuhl Seminar Proceedings 07122
Normative Multi-agent Systems
http://drops.dagstuhl.de/opus/volltexte/2007/937

# 1 Motivation for a new approach to normative systems

In Section 1.1 we explain why we need a theory of norms by arguing that norms are a special class of constraints deserving special analysis. In Section 1.2 we define what we mean by a norm, distinguishing among regulative, constitutive and procedural ones, and in Section 1.3 we explain why a normative system in multi-agent systems is seen as a mechanism, in particular to obtain desirable agent behavior or to structure organizations. Finally we explain in Section 1.4 what we mean by game-theoretic scenarios in normative multi-agent systems, and in Section 1.5 we discuss an important advantage of our game-theoretic approach, which we call the game-theoretic analysis of normative multi-agent systems.

## 1.1 Norms are a class of constraints deserving special analysis

Meyer and Wieringa define normative systems as "systems in the behavior of which norms play a role and which need normative concepts in order to be described or specified" [100, preface]. Alchourròn and Bulygin [2] define a normative system inspired by Tarskian deductive systems:

> "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 $\alpha$ there is at least one normative correlation, we shall say that the set $\alpha$ has normative consequences. A system of sentences which has some normative consequences will be called a normative system." [2, p.55].

Jones and Carmo [89] introduce agents in the definition of a normative system by defining it as "sets of agents whose interactions are 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." Since the agents' control over the norms is not explicit here, we use the following definition.

> A normative multi-agent system is a multi-agent system together with normative systems in which agents can decide whether to follow the explicitly represented norms or not, and the normative systems specify how and in which extent the agents can modify the norms. [68]

Note that this definition makes no presumptions about the internal architecture of an agent or of the way norms find their expression in agent's behavior.

**Representation of norms** Since norms are explicitly represented, according to our definition of a normative multi-agent system, the question should be raised how norms are represented. Norms can be interpreted as a special kind of constraint, and represented depending on the domain in which they occur.

However, the representation of norms by domain dependent constraints runs into the question what happens when norms are violated. Not all agents behave according to the norm, and the system has to deal with it. In other words, norms are not hard constraints, but soft constraints. For example, the system may sanction violations or reward good behavior. Thus, the normative system has to monitor the behavior of agents and enforce the sanctions. Also, when norms are represented as domain dependent constraints, the question will be raised how to represent permissive norms, and how they relate to obligations. Whereas obligations and prohibitions can be represented as constraints, this does not seem to hold for permissions. For example, how to represent the permission to access a resource under an access control system? Finally, when norms are represented as domain dependent constraints, the question can be raised how norms evolve.

We therefore believe that norms should be represented as a domain independent theory. For example, deontic logic [94,95,96,109,110,117] studies logical relations among obligations and permissions, and more in particular violations and contrary-to-duty obligations, permissions and their relation to obligations, and the dynamics of obligations over time. Therefore, insights from deontic logic can be used to represent and reason with norms in multi-agent systems. Deontic logic also offers representations of norms as rules or conditionals. However, there are several aspects of norms which are not covered by constraints nor by deontic logic, such as the relation among the cognitive abilities of agents and the global properties of norms. Meyer and Wieringa explain why normative systems are intimately related with deontic logic.

> "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! Deontic logic provides a means to do just this by using special modal operators that indicate the status of behavior: that is whether it is legal (normative) or not" [100, preface].

**Norms and agents** Conte *et al.* [76] distinguish two distinct sets of problems in normative multi-agent systems research. On the one hand, they claim that legal theory and deontic logic supply a theory of norm-governed interaction of autonomous agents while at the same time lacking a model that integrates the different social and normative concepts of this theory. On the other hand, they claim that three other problems are of interest in multi-agent systems research on norms: how agents can acquire norms, how agents can violate norms, and how an agent can be autonomous. Agent decision making in normative systems and the relation between desires and obligations has been studied in agent architectures [72], which thus explain how norms and obligations influence agent behavior.

An important question in normative multi-agent systems is where norms come from. Norms are not necessarily created by legislators, but they can also be negotiated among agents, or they can emerge spontaneously, making the agents norm autonomous [112]. In electronic commerce research, for example, cognitive foundations of social norms and contracts are studied [58]. Protocols and social mechanisms are now being developed to support such creations of norms in multi-agent systems. Moreover, agents like legislators playing a role in the normative system have to be regulated themselves by procedural norms [67], raising the question how these new kind of norms are related to the other kinds of norms.

When norms are created, the question can be raised how they are enforced. For example, when a contract is violated, the violator may have to pay a penalty. But then there has to be a monitoring and sanctioning system, for example police agents in an electronic institution. Such protocols or roles in a multi-agent system are part of the construction of social reality, and Searle [105] has argued that such social realities are constructed by constitutive norms. This raises the question how to represent such constitutive or counts-as norms, and how they are related to regulative norms like obligations and permissions [62].

**Norms and other concepts** Not only the relation between norms and agents must be studied, but also the relation between norms and other social and legal concepts. How do norms structure organizations? How do norms coordinate groups and societies? How about the contract frames in which contracts live? How about the relation between legal courts? Though in some normative multi-agent systems there is only a single normative system, there can also be several of them, raising the question how normative systems interact. For example, in a virtual community of resource providers each provider may have its own normative system, which raises the question how one system can authorize access in another system, or how global policies can be defined to regulate these local policies [62].

### 1.2 Kinds of norms

Normative multiagent systems as a research area can be defined as the intersection of normative systems and multi-agent systems [68]. With 'normative' we mean 'conforming to or based on norms', as in *normative behavior* or *normative judgments*. According to the Merriam-Webster Online Dictionary [99], 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. Kinds of norms which are usually distinguished are regulative norms

like obligations, permissions and prohibitions, constitutive norms like counts-as conditionals, and more, as discussed below.

**Regulative norms: obligations, permissions, prohibitions** Regulative norms specify the ideal and varying degrees of sub-ideal behavior of a system by means of obligations, prohibitions and permissions. Deontic logic [6,118] considers logical relations among obligations and permissions and focuses on the description of the ideal or optimal situation to achieve, driven by representation problems expressed by the so-called deontic paradoxes, most notoriously the contrary-to-duty paradoxes, see, for example, [89,110].

**Constitutive norms: counts-as conditionals** 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 [105] has been interpreted in deontic logic in different ways and it seems to refer to different albeit related phenomena [85]. For example, Jones and Sergot [90] consider counts-as from the constitutive point of view. According to Jones and Sergot, the fact that A counts-as B in context C is read as a statement to the effect that A represents conditions for guaranteeing the applicability of particular classificatory categories. The counts-as guarantees the soundness of that inference, and enables "new" classifications which would otherwise not hold.

An alternative view of the counts-as relation is proposed by Grossi *et al.* [84]: according to the classificatory perspective A counts-as B in context C is interpreted as: A is classified as B in context C. In other words, the occurrence of A is a sufficient condition, in context C, for the occurrence of B. Via counts-as statements, normative systems can establish the ontology they use in order to distribute obligations, rights, prohibitions, permissions, *etc.* See [54] for a discussion on the relation between count-as conditionals, classification and context.

In [42,52,58] we propose a different view of counts-as which focuses on the fact that counts-as often provides an abstraction mechanism in terms of institutional facts, allowing the regulative rules to refer to legal notions which abstract from details. Counts-as conditionals can be used to define other concepts, such as role-based Rights in Artificial Social Systems [50]. In [51,60] we study the relation between obligations, permissions and constitutive norms using a logical architecture.

**Procedural norms** The distinction between substantive and procedural norms is well known in legal theory [98]. 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. Procedural law encompasses legal rules governing the process for settlement of disputes (criminal and civil). Procedural and substantive law are complementary. Procedural law brings substantive law to life and enables rights and duties to be enforced and defended. For example, procedural norms explain how a trial should be carried out and which are the duties, rights and powers of judges, lawyers and defendants.

The role that agents have in enforcing the social order the normative system aims to by creating norms has been recognized in normative multi-agent systems [58,62], and agents are considered which are in charge of sanctioning violations on behalf of the normative system [33,39]. Moreover, obligations are associated with procedural norms which are instrumental - to use Hart [86]'s terminology - to distribute the tasks to agents like judges and policemen, who have to decide whether and how to fulfill them.

In [55,67] we introduce a logical framework for substantive and procedural norms, and we use it to study the relation between these two kinds of norms and to answer the following three questions. First, how are regulative and constitutive norms related in a normative system with substantive and procedural norms? Second, by which mechanism are procedural norms created to motivate agents to recognize violations, apply sanctions, or to recognize institutional facts? Third, how can the formal framework be used to model various applications of normative multi-agent systems, where only some of them may need procedural norms?

### 1.3 Normative system as a mechanism

In this section we discuss why there are norms in social systems like multi-agent systems. We have distinguished various kinds of norms, such as obligations or counts-as conditionals, but this does not explain their existence. We assume that a norm is a mechanism to obtain desired multi-agent system behavior. In other words, it is an incentive, which brings us directly into the study of incentives, called economics.

**Norms as a mechanism to obtain desirable agent behavior** Norms have for long been considered as one of the possible incentives to motivate agents. Consider the economist Levitt [92, p.18-20], discussing an example of Gneezy and Rustichini [81].

> Imagine for a moment that you are the manager of a day-care center. You have a clearly stated policy that children are supposed to be picked up by 4 p.m. But very often parents are late. The result: at day's end, you have some anxious children and at least a teacher must wait around for the parents to arrive. What to do?
> A pair of economists who heard of this dilemma – it turned out to be a rather common one – offered a solution: fine the tardy parents. Why, after all, should the day-care center take care of these kids for free?

> The economists decided to test their solution by conducting a study of ten day-care centers in Haifa, Israel. The study lasted twenty weeks, but the fine was not introduced immediately. For the first four weeks, the economists simply kept track of the number of participants who came late; there were, on average, eight pickups per week per day-center. In the fifth week, the fine was enacted. It was announced that any parent arriving more than ten minutes late would pay $3 per child for each incident. The fee would be added to the parents' monthly bill, which was roughly $380.
> After the fine was enacted, the number of late pickups promptly went ... up. Before long there were twenty late pickups per week, more than double the original average. The incentive had plainly backfired.
> 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.

The daycare example illustrates that norms can be used as a mechanism to obtain desirable behavior of a multiagent system, because it is used as one of the incentives. It suggests also that the main tools to study incentives in economics, classical decision and game theory, may be useful tools to study the role of normative incentives too. Note that the daycare example illustrates also that economic theory is concerned with normative reasoning too, and that an analysis of incentives should not naively restrict itself to economic incentives, because it should also take norms into account.

The fact that norms can be used as an 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 [87] for a discussion.

**Norms as a mechanism 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.

According to Zambonelli *et al.* [119] "a multiagent system can be conceived in terms of an organized society of individuals in which each agent plays specific roles and interacts with other agents". At the same time, they claim that "an organization is more than simply a collection of roles (as most methodologies assume) [...] further organization-oriented abstractions need to be devised and placed in the context of a methodology [...] As soon as the complexity increases, modularity and encapsulation principles suggest dividing the system into different suborganizations". According to Jennings [88], however, most current approaches "possess insufficient mechanisms for dealing with organisational structure". Moreover, what is the semantic principle which allows decomposing organizations into suborganizations must be still made precise. Organizations are modelled as collections of agents, gathered in groups [78], playing roles [88,97] or regulated by organizational rules [119].

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 [65]. 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 interdependences 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 [104] 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".

### 1.4 Game-theoretic scenarios of normative multiagent systems

In this section we explain our game-theoretic foundations for norms [59].

**Interactions among agents in a normative multiagent system** Many examples are given in the literature on the interaction among agents using regulative norms. For example, consider the following simple scenario due to Ron Lee of the access to a photo copier [91]:

1. A tells B to permit C to do use photocopier
2. B permits C to do use photocopier
3. C cannot do use copier, since door is closed
4. A complains to B about C not able to use it
5. B tells A that he permitted C, as requested

The standard analysis of this example includes the idea that C has the right to use the photocopies in the sense that he is entitled to use it, and B is therefore obliged to open the door. Moreover, as the photocopier has an access code, then B has to tell the code to C (even though C would be able to use the copier if he where to guess the code, which is the point where knowledge gets into this scenario). There are many variants of this example due to Sergot and colleagues, such as borrowing books in the library regulations formalized, parking cars in the parking lot in the parking regulations, and so on. Typically, many more agents are involved in these real world examples than just A, B and C. In multiagent systems, similar scenarios can be found in access control systems, for example to access a web service.

A similar example is often discussed where permission is replaced by obligation (where A like to transmit its will to influence the behavior of C via B):

1. A tells B to oblige C to do copies of a paper.
2. B obliges C to do copies of a paper
3. C does not do copies of a paper
4. A complains to B about C not doing copies of a paper
5. B tells A that he obliged C, as requested

Contracts are based on norms and occur in strategic interaction scenarios found in e-commerce, as studied by Tan and colleagues [83]. In an escrow service or a bill of lading typically several buyers, sellers, transporters, financial institutions and other agents are involved, which regulate their interactions via complex contracts. Here norms are used to give agents the power to achieve things.

When more agents are involved, their social interactions may give rise to the emergence of norms. For example, in case there is no trust there is no deal due to lack of equilibrium. If there is a joint goal based on agent desires then the agents can propose or negotiate norms leading to a new equilibrium, in which they accept the norms.

In most human social systems it takes a long time to emerge, but in computer systems they can be created much more quickly. For example, consider a peer to peer ad hoc network used for incident management. In case of an incident such as fire in a tunnel, cars, police, firemen, hospitals and so on have to be coordinated.

Another source for social interaction scenarios can be found in the popular reality games such as second life, where we expect many applications of normative multiagent systems.

**Complexity and abstraction** All these examples of interaction scenarios show highly complex and dynamic systems, and the question is how we can model the examples - whether it is to develop multi-agent systems in agent based software engineering, or to analyze the multi-agent system in agent theory. There are two main approaches to reduce the complexity.

First, the usual approach to reduce the complexity is to describe agents using a simple and uniform formalization. For example, classical game theory describes all agents by utility function and probability distribution, together with the decision rule to maximize expected utility, and alternative agent models developed in artificial intelligence and cognitive science are based on models such as belief-intention-desire (BDI) model.

Second, an alternative approach to reduce the complexity is to restrict the number of agents which are considered in the interaction. Here the multi-agent structure of normative systems can be used. For example, legal systems are based on the Trias Politica [39]. From the perspective of a normative system, there are two kinds of agents. First, the agents who are subject to the norms, and the agents who play a role in the system to make it function. This generates to the distinction between substantive norms used to regulate agents subject to the system, and procedural norms used to regulate agents playing a role in the system.

**Normative system as a level of abstraction** One way we can simplify interaction in a normative multiagent system is to abstract away the agents playing a role in the normative system, and keeping the normative system as an entity

interacting with the agents subject to it. The agents playing a role in the system empower the normative system, and the normative system delegates again some of its powers to these agents; this is known as *mutual empowerment*. This level of abstraction is most clear in organizational theory, where an organization can be seen as a normative multiagent system as well as a legal entity. One can use the agent metaphor for such abstracted normative systems, for example by attributing mental attitudes to normative systems [32].

The sociologist Goffman sees norms as producing a form of strategic interaction between the agent and the normative system. In a normative system, the "enforcement power is taken from mother nature and invested in a social office specialized for this purpose, namely a body of officials empowered to make final judgements and to institute payments" [82, p.115]. Such a game is unusual since "the judges and their actions will not be fully fixed in the environment, many unnatural things are possible. [...] the payment for a player's move ceases to be automatic but is decided on and made by the judges where everything is over" [82, p.115]. "Strategic interaction" here means the, according to Goffman unavoidable, taking into consideration of the other agents' actions.

> "When an agent considers which course of action to follow, before he takes a decision, he depicts in his mind the consequences of his action for the other involved agents, their likely reaction, and the influence of this reaction on his own welfare" [82, p. 12].

At this level of abstraction, the simplest game which can be played is an agent deliberating about an action, and the normative system reacting to it. Since the goal of the agent is typically to violate the norms without being sanctioned, we call this kind of interaction a *violation game*, and we represent it by A:N. Other kinds of interactions at this abstraction level are extensions of violation games. consider for example a (legislator in a) normative system deliberating which norm to create. He can introduce a norm, then an agent will play a violation game. The goal of the normative system is that the agent is motivated such that the norm is not violated. We call it a norm creation game, and we abstractly represent it by N:A:N. Also more complex interactions among agents can be modelled in this way, for example involving control hierarchies such as defender agents, various kinds of authorities like norm source hierarchies, and so on.

### 1.5 The game-theoretic analysis of norms

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, otherwise the norms would loose their force to motivate agents. Moreover, sanctions should not be too low, as in the daycare example, but they also should not be too high, as shown by the argument of Beccaria [14] on death penalty. Otherwise, once a norm is violated, there is no way to prevent further norm violations. In [59] we list the following requirements for such an analysis.

The first requirement is that norms influence the behavior of agents. However, they only have to do so under normal or typical circumstances. For example, if other agents are not obeying the norm, then we cannot expect an agent to do so. This norm acceptance has been studied by [76], and in a game-theoretic setting for social laws by [108].

The second requirement is that even if a norm is accepted in the sense that the other agents obey the norm, an agent should be able to violate the norms. A normative multi-agent system is a "set of agents [...] whose interactions can 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" [89]. In other words, the norms of global policies must 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. Moreover, detective control is the result of actions of agents and therefore subject to errors and influenceable by actions of other agents. Therefore, 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. A game-theoretic analysis can be used to study these issues of fraud and deception.

The third requirement is that norms should apply to a variety of agent types, since agents can be motivated in various ways, as the daycare example illustrates. We assume that a norm is a mechanism to obtain desired multi-agent system behavior, and must therefore under normal or typical circumstances be fulfilled for a range of agent types. Castelfranchi argues that sanctions are only one of the means which motivate agents to respect obligations, besides "pro-active actions, prevention from deviation and reinforcement of correct behavior, and then also 'positive sanctions', social approval" [74]. Castelfranchi [74] argues that an agent should fulfill an obligation because it is an obligation, not because there is a sanction associated with it.

> "True norms are aimed in fact at the internal control by the addressee itself as a cognitive deliberative agent, able to understand a norm as such and adopt it. [...] The use of external control and sanction is only a sub-ideal situation and obligation." [74]

We therefore use the distinction between violations and sanctions to distinguish between the agent's interpretation of the obligation, and its personal characteristics or agent type. The agent types are inspired by the use of agent types in the goal generation components of Broersen et al.'s BOID architecture [73]. *Roughly*, we distinguish among *norm internalizing agents*, *respectful* agents that attempt to evade norm violations and that are motivated by what counts and

does not count as a violation, and *selfish* agents that obey norms only due to the associated sanctions, i.e. that are motivated by sanctions only. An obligation without a sanction *should* be fulfilled, as Castelfranchi argues. But if fulfilling the obligations has a cost then it *is* only fulfilled by respectful agents, not by selfish agents, unless some incentives are provided or the agents dislike some social consequences of the violations. A respectful agent fulfills its obligations due to the existence of the obligation, whereas a selfish agent fulfills its obligations due to fear of consequences.

**Respectful agents:** agents that base their decisions solely on whether their behavior respects the goals of the normative agents. They put their duties before their own goals and desires: they maximize the fulfilment of obligations regardless to what happens to its own goals; even if the agent **n** did not sanction them, the agent **a** would prefer to respect the obligation. We say that respectful agents *adopt* the goal of the normative agent as their preference.

**Selfish agents:** agents that base their decisions solely on the consequences of their actions. If the obligation is respected, it is because agent **a** predicts that the situation resulting from the fulfillment is preferred according to its own goals and desires only: e.g., if it does not share its files, it knows that it can be sanctioned, a situation it does not desire or want. But it is possible also that there are not only material reasons, that is, not only for the damage caused by the sanction. Nothing prevents that the content of the norm is already a goal of the agent; moreover, agent **a** could have the desire not to be considered a violator, or it knows that being considered a violator gives it a bad reputation, so that it would not be trusted by other agent. However, to stick to the obligation, the goal of not being a violator or of not being sanctioned must be preferred by the agent to the desire or goal not to respect the obligation (obligations usually have a cost): a weak sanction, as it often happens, does not enforce the respect of a norm (e.g., the sanction is that the access to a website is forbidden, but the agent has already downloaded what it wanted).

To distinguish these cases, we distinguish between the decision to count behavior as a violation, and to sanction it.

Of course, most agents are mixed types of agents between these two extremes. Sometimes an agent is respectful and in other cases it is selfish. Balancing these two extremes is an important part of the agent's deliberation. The adoption of the obligation as a goal can be considered as an additional factor when the different alternatives are weighed according to its own goals and desires: so that the newly added motivations can affect the decision of agent **a** and move it towards an obligation-abiding behavior besides its own attitude towards that goal and the possible consequences of its alternative decisions. However, if a norm is effective in each case of the agent types, it is also effective for mixed agents. therefore we can restrict ourselves to the extreme agent types in a game-theoretical analysis.

Given possible conditions for a norm, the fourth requirement is that norms are as weak as possible, in the sense that the norms should not apply in cases where this is undesired, and that sanctions should not be too severe. The latter is motivated by a classical economic argument due to Beccaria, which says that if sanctions are too high, they can no longer be used in cases where agents already have violated a norm. Sanctions should be high enough to motivate selfish agents, but they should not be too high.

Designing norms satisfying these requirements is an area of game theory called mechanism design. In, amongst others, [59] we provide the following informal definition of obligation, extending Boella and Lesmo [26]'s proposal. According to legal studies what distinguishes norms from mere power to damage an agent is that sanctions are possible only in case of violations and which situations can be considered as violations is defined by the law: "*nullum crimen, nulla poena sine lege*". In our definition a norm specifies what will be considered as a violation by the normative agent (item 2) and that the normative agent will sanction only in case of violations (item 3). In this paper, we consider the set of norms as given. Given a set of norms $N$, agent **a** is obliged by the normative agent **n** to do $x$ with sanction $s$, iff there is a norm $n$ of $N$ such that:

- The content $x$ of the obligation is a desire and goal of **n** and agent **n** wants that agent **a** adopts this as its decision since it considers agent **a** as responsible for $x$.
- agent **n** has the desire and the goal that, if the obligation is not respected by agent **a**, a prosecution process is started to determine if the situation "counts as" a violation of the obligation and that, if a violation is recognized, agent **a** is sanctioned.
- Both agent **a** and agent **n** do not desire the sanction: for agent **a** the sanction is an incentive to respect the obligation, while agent **n** has no immediate advantage from sanctioning.

This definition is extended in various papers in a number of ways. For example, goals and desires are formalized as conditional rules, because norms and obligations are typically represented by conditional rules.

## 2 Objectives

In this section we discuss the four objectives of our work on game-theoretical approach to normative multiagent systems.

### 2.1 A representation of a normative multiagent systems that combines the three existing representations of normative multiagent systems

There are many approaches to conceptualizing and developing normative multiagent systems. The most popular class of approaches starts from logical relations

among obligations (and sometimes permissions) in a deontic logic, and then extends the formalism with agent concepts like actions and time. Since the norms in normative multiagent systems are typically represented explicitly, we may say that these approaches start from the representation of norms. We call it the deontic logic approach. Drawback of this approach is that there is no guideline to tell how norms affect behavior.

The second class of approaches in normative multiagent systems starts from the use of norms in agent decision making and interaction. In other words, given a set of norms, how do the agents behave? We call it the normative agent architecture approach. This more dynamic approach, focussing on behavior rather than normative system structure, has the drawback that it does not give a guideline how the normative system changes over time due to behavior of agents.

The third class of approaches in normative multiagent systems takes the strategic interaction among agents and (representatives of) the normative system as a starting point. We call it the game-theoretic approach. An example is the distinction between controllable and uncontrollable agents in the Tennenholtz and Brafman's model. A drawback of their quantitative model is that it is not easily combined with the other two approaches, such that explicit representation of norms and normative decision making remains a problem.

Our first objective is to build a game theoretic model of normative multiagent systems that extends both the deontic logic and the normative decision making approach. We therefore refer to Goffman's notion of strategic interaction rather than the dominant economic equilibrium analysis. Particular challenges are:

**KR-ASS** Combining the qualitative formalisms used in knowledge representation and the quantitative ones used in artificial social systems,

**AA-ASS** Combining the micro representation of agent architectures with the macro representations in social theories (the micro-macro dichotomy)

**KR-AA** The use of logic in knowledge representation and the use of architecture in agent theory.

Typically there are several ways to align two theories in a common framework. For example, for the latter problem, we may develop logical representations of agents [73], or we may develop an architecture for a normative system [60].

### 2.2 A logical framework for qualitative risk analysis, building on ideas in security and risk management

Risk analysis goes beyond traditional security by not only stating what is forbidden, but also accepting that things go wrong sometimes, and how to deal with them. Therefore, risk analysis not only needs constraints like security, but also contrary-to-duty or more generally normative reasoning.

Traditional risk analysis is quantitative and based on statistics. However, it does not take the organizational structure, legal consequences and so on into account. If we model a system, we may also take a more qualitative approach. What we have to do to analyze risk is to build normative systems and agent

models, and combine them. This is precisely what we do in our normative multiagent systems. Thus, we replace classical statistical risk analysis by our game theoretic analysis.

This thus explains why contrary-to-duty reasoning is essential for risk management, but there is much more to risk management than contrary-to-duty reasoning. In particular, given that agents can violate norms and can be sanctioned, will agents violate the norms? Moreover, in such cases, will they be sanctioned?

### 2.3 A classification of situations in which one needs which elements of normative systems as a social mechanism

In Section 1.2 we discussed various kinds of norms such as obligations, permissions, counts-as conditionals, and procedural norms. Moreover, there are social laws, conventions, rights, entitlements, legal institutions, and many more related concepts. Each of these concepts may be considered as another kind of mechanism. At this moment there is no consensus when we need these mechanisms, and when we can use a simpler normative multiagent systems. Thus far only two kinds of arguments: obligations are needed when there is contrary-to-duty reasoning [89], and permissions are needed for multiple authorities [2].

### 2.4 Examples and classification which kind of normative multiagent systems are to be used for which kind of applications in computer science

Many new technologies use essentially the same kind of normative multiagent system as older ones, but each application domain tends to reinvent its own normative system, typically in a very naive way. For example, consider the use of rollback and compensation in web technology, which may be seen as preventative and detective control systems. We therefore aim to illustrate each kind of normative multiagent system by a typical example, such as fraud and deception, electronic commerce, secure knowledge management, and so on.

### 2.5 Scope of the objectives

We consider the organizational structures with its explicit roles as an orthogonal issue to the game-theoretic approach to normative multi-agent systems. In other words, we can study normative multi-agent systems without making these aspects explicit. We discuss some of our results in this area in Section 4.6.

We do not discuss the design or implementation of normative multiagent systems, though we believe that our programming language powerJava may be used to implement normative multiagent systems.

Though norms are used to control the emergent behavior in MAS, and evolutionary game theory is a useful tool to study this emergence, we do not address this issue.

## 3 Methodology

In our approach we use input/output logic for deontic logic, BOID architecture for the agent architecture, and both game theory and recursive modeling for artificial social systems.

### 3.1 Input/output logic

Input/output logic takes its origin in the study of conditional norms. These may express desired features of a situation, obligations under some legal, moral or practical code, goals, contingency plans, advice, etc. Typically they may be expressed in terms like: *In such-and-such a situation, so-and-so should be the case*, or *... should be brought about*, or *... should be worked towards*, or *... should be followed* – these locutions corresponding roughly to the kinds of norm mentioned.

### 3.2 BOID Architecture

The BOID architecture [73] is an extension of the BDI architecture with obligations (O). Each mental attitude is represented by a component in the architecture, whose behavior is described by input/output logic. Moreover, qualitative decision theories have been developed which extend this rule based formalism with the decision rule to achieve goals and to evade goal violations.

In the BOID architecture there is no set of norms and norm descriptions, but instead the agent description is adapted such that obligations (O) are added to the mental states of agents. This can be interpreted as a kind of internalization of the normative system by the agents, or as an abstraction which abstracts away the normative system. This is the dominant approach in deontic logic [100], in which typically one abstracts away from the norms to study logical relations between obligations (though for criticism and alternative approaches see [1,116]). Alternatively, in approaches based on the so-called Anderson reduction [4,5], which defines obligation of $p$ as the necessity that the absence of $p$ leads to a violation, $O(p) = \Box(\neg p \rightarrow V)$, obligations are defined in terms of violability and the state of the world. In the variant proposed by Meyer [101], who defines obligation of action $\alpha$ as 'the absence of $\alpha$ leads to a violation state, or $O(\alpha) = [\overline{\alpha}]V$, obligation is defined in terms of the agent's behavior.

### 3.3 Tennenholtz' classical game-theoretic approach to artificial social systems

We first consider the so-called partially controlled multi-agent system (PCMAS) approach of Brafman and Tennenholtz [70], one of the classical game-theoretical studies of social laws in so-called artificial social systems developed by Tennenholtz and colleagues, because incentives like sanctions and rewards play a central role in this theory. So-called controllable agents – agents controlled by the system programmer – enforce social behavior by punishing and rewarding agents,

and thus can be seen as representatives of the normative system. For example, consider an iterative prisoner dilemma. A controlled agent can be programmed such that it defects when it happens to encounter an agent which has defected in a previous round.

The PCMAS model thus distinguishes between two kinds of agent interaction in the game theory, namely between two normal (so-called uncontrollable) agents, and between a normal and a controllable agent. We show in this paper that this makes it a very useful model to give game-theoretic foundations to norms. Whereas classical game theory is only concerned with interaction among normal agents, it is the interaction among normal and controllable agents which we use in our game theoretic foundations.

The PCMAS approach not only clarifies the design of punishments, but it also illustrates the iterative and multi-agent character of social laws. However, there are also drawbacks of the model, such that it cannot be used to give a completely satisfactory game-theoretic foundation for norms. We would like to express that a norm can be used for various kinds of agents, such as norm internalizing agents, respectful agents that attempt to evade norm violations, and selfish agents that obey norms only due to the associated sanctions. Therefore, as classical game theory is too abstract to satisfactorily distinguish among agent types, we consider also cognitive agents and qualitative game theory.

Several game-theoretic studies on social laws have been made by Tennenholtz and colleagues, for example based on off-line design of social laws [106], the emergence of conventions [107], and the stability of social laws [108]. The approach of Brafman and Tennenholtz [70] distinguishes between controllable and uncontrollable agents, analogous to the distinction between controllable and uncontrollable events in discrete event systems.

Controllable agents are agents controlled by the system programmer to enforce social behavior by punishing and rewarding agents. The game-theoretic model is the most common model for representing emergent behavior in a population. A single game consists of the usual payoff matrix. For example, the prisoner's dilemma is a two person game where each agent can either cooperate or defect.

**Definition 1.** *A k-person game g is defined by a k-dimensional matrix M of size $n_1 \times \ldots \times n_k$, where $n_m$ is the number of possible actions (or strategies) of the m'th agent. The entries of M are vectors of length k of real numbers, called pay-off vectors. A joint strategy in M is a tuple $(i_1, i_2, \ldots, i_k)$, where for each $i \leq j \leq k$, it is the case that $1 \leq i_j \leq n_j$.*

An iterative game consists of a sequence of single games.

**Definition 2.** *A n-k-g iterative game consists of a set of n agents and a given k person game g. The game is played repetitively an unbounded number of times. At each iteration, a random k-tuple of agents play an instance of the game, where the members of this k-tuple are selected with uniform distribution from the set of agents.*

Efficiency is a global criterion for judging the "goodness" of outcomes from the system's perspective, unlike single payoffs which describe a single agent's perspective.

**Definition 3.** *A joint strategy of a game $g$ is called* efficient *if the sum of the players pay-offs is maximal.*

New in the Brafman-Tennenholtz model are the notions of punishment and reward w.r.t. some joint strategy $s$, measuring the gain (benefit) or loss (punishment) of an agent if we can somehow change the joint behavior of the agents from a chosen efficient solution $s$ to $s'$.

**Definition 4.** *Let $s$ be a fixed joint strategy for a given game $g$, with pay-off $p_i(s)$ for player $i$; in an instance of $g$ in which a joint strategy $s'$ was played, if $p_i(s) \geq p_i(s')$ we say that $i$'s* punishment *w.r.t. $s$ is $p_i(s) - p_i(s')$, and otherwise we say that its* benefit *w.r.t. $s$ is $p_i(s') - p_i(s)$.*

Agents may need to be constrained to behave in a way that is locally suboptimal such that the multi-agent system is as efficient as possible. Brafman and Tennenholtz call such a constraint a social law. Then they informally define controlled agents:

> "Agents not conforming to the social law are referred to as *malicious agents*. In order to prevent the temptation to exploit the social law, we introduce a number of *punishing agents*, designed by the initial designer, that will play 'irrationally' if they detect behavior not conforming to the social law, attempting to minimize the payoff of malicious agents. The knowledge that future participants have of the punishment policy would deter deviations and eliminate the need for carrying it out. Hence, the punishing behavior is used as a threat aimed at deterring other agents from violating the social law. This threat is (part of) the control strategy adopted by the controllable agents in order to influence the behavior of the uncontrollable agents. Notice that this control strategy relies on the structural assumption that uncontrollable agents are expected utility maximizers."

They consider the design of punishments, and show, for example, necessary and sufficient conditions for the existence of a punishing strategy.

We believe that PCMAS can be used to give game-theoretic foundations to norms, though Brafman and Tennenholtz do not use or consider the terminology of normative systems or deontic logic. The model fulfills our two requirements by explaining several aspects of norms, such as the fact that they can be used iteratively, that sanctions are associated to it, and that they can be applied to various kinds of agents.

In particular, a useful property of the PCMAS model is that it uses the game-theoretic machinery to study not only interaction among normal agents, but also interaction among the controlled agents and the normal agents. Since

the controlled agents are representatives of the normative system, this means that the game-theoretic machinery is used to study the interaction among the normative system and the agents.

However, the emphasis on modeling uncontrollable agents as utility maximizers implies that they only obey the norm because they are afraid of the sanction. Thus the model does not fulfill the third requirement because it seems to exclude the possibility that an agent obeys the norm simply due to its existence. In social theory, for example, agents have been studied which internalize norms in the sense that they incorporate norms as their own goal, or respectful agents trying to obey the norms without internalizing them.

Maybe the game-theoretic machinery can be extended to take such social agents into account. For example, a norm internalizing agent may be defined as an uncontrollable agent which simply copies the utility function of a punishing agent, and a respectful agent which avoids sanctions even when the number of punishing agents is too low, for example by assuming the number of punishing agents is much higher than it is in reality. They may for example be ashamed to be caught while driving without a train ticket.

However, such a solution does not seem very satisfactory. For the norm internalizing agents, they not only obey the norm but they also start to act as policemen, which seems to go to far. Moreover, even when punishment is low or absent, a respectful agent may obey the norm (as in the daycare example). There seem to be several alternative ways to define respectful agents, but they seem to have their own drawbacks.

Moreover, there are also some more technical problems. For PCMAS to give game-theoretic foundations to norms, we first have to define the syntax of a norm. Typically norms are expressed as modal sentences expressing that $p$ is obliged, $O(p)$ in a deontic logic, or $p$ is permitted, $P(p)$. Since in the PCMAS setting we have actions or strategies only, we define $O_i(\alpha, p)$ for agent $i$ is obliged to do action $\alpha$, otherwise he is sanctioned with punishment $p$. Since a punishment $p$ is defined as $p_i(s) - p_i(s')$, the first problem is how to define the chosen efficient solution $s'$. It is implicit in the condition that in the situation in which no norm is violated, no agent is punished (the Nero/Caligula example of the introduction).

Finally, whether an obligation $O_i(\alpha, p)$ holds in PCMAS or not cannot be seen from the game's definition, but only from the behavior of the controlled agents. In other words, it can only be derived from the design of punishments not explicit in the game theory.

### 3.4 Recursive modeling

Classical decision and game theory have been criticized for their assumptions of ideality. Several alternatives have been proposed that take the limited or bounded rationality of decision makers into account. For example, Newell [102] 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. Brat-

man [71] further extends such theories with intentions for sequential decisions and norms for multiagent decision making.

Alternatively, Gmytrasiewitcz and Durfee [80] 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.

> "Recursive modelling method views a multi agent situation from the perspective of an agent that is individually trying to decide what physical and/or communicative actions it should take right now." [80]

Boella and Lesmo [26] therefore propose the following definition of a sanction-based obligation in terms of beliefs, goals and desires, inspired by Goffman's game-theoretic interpretation of obligations and by recursive modelling. Boella and Lesmo formalize this definition in a game-theoretic framework in which they recursively model the normative agent's behavior. The formalization is based on multi-attribute utility theory for taking into account the different aspects of the world for which agents have preferences. An important advantage of Boella and Lesmo's definition is that it does not introduce additional mental attitudes, but it defines obligations in terms of beliefs, desires and goals. Moreover, they distinguish various reasons why agents fulfil or violate obligations.

> "An obligation holds when there is an agent A, the *normative* agent, who has a goal that another (or more than one) agent B, the *bearer* agent, satisfies a goal G and who, in case he knows that the agent B has not adopted the goal G, can decide to perform an action Act which (negatively) affects some aspect of the world which (presumably) interests B. Both agents know these facts." [26, p.496]

The approach overcomes some of the limitations discussed in the previous section. First, obligations of the agents can be formalized as desires or goals of the normative agent. This representation may be paraphrased as "Your wish is my command", the title of this paper, because the desires or wishes of the normative agent are the obligations or commands of the other agents. The goals of the normative system describe the ideal behavior of the system.

Second, structural relations between agents playing roles in a normative system like legislators, who create norms, judges, who count behavior as violations and associate sanctions, and policemen who enforce sanctions, can be represented by the standard hierarchical structure of agents. For example, the normative agent $a_{n+1}$ may contain the role of a legislator $a_1$, a judge $a_2$ and a policeman $a_3$. Such a relation between the normative agent and other agents is probably not reductive, as the normative system also contains properties which cannot be reduced to roles of normal agents.

Third, as illustrated by Boella and Lesmo by several examples, agents take the normative system into account by playing games with it. For example, an agent considers whether its actions will lead to a reaction of the normative system such as being sanctioned. In their model, the agent can evade sanctions by

for example ensuring that the normative agent does not observe their behavior, or by bribing the system. The advantage of the approach is that standard techniques developed in decision and game theory can be applied to normative reasoning. Moreover, a legislator can play a game with the normative agent and another agent to see whether a new norm it introduces will be complied with, and which kind of sanctions it has to associate with the norm to achieve the desired behavior.

The normative systems as agents perspective together with the attribution of mental attitudes to these normative systems is not only, as Tuomela [111] proposed, a powerful metaphor leading to useful techniques, but it can also be explained from a conceptual point of view in several ways. For example, according to Wooldridge and Jennings [115] the conditions for calling a system an agent are its autonomy to control its actions and internal state, its social ability to interact with other agents, its reactivity to the changes it observes in the environment, and its pro-activeness due to goal directed behavior and taking the initiative. All these conditions are met by normative systems. First, it is autonomous in counting behavior as violations and applying sanctions. Second, it interacts with these other agents by influencing their behavior, and the other agents interact with it by for example creating norms. Third, it is reactive since counting behavior as a violation of norms depends on its observations of such behavior. Fourth, it is proactive since this action is taken without any explicit triggering action of the agent, who would of course instead try to evade being violators and being sanctioned.

Moreover, attribution of mental attitudes to normative systems can be explained by the interpretation of normative multiagent systems as dynamic social orders. According to Castelfranchi [74], a social order is a pattern of interactions among interfering agents "such that it allows the satisfaction of the interests of some agent A". These interests can be a shared goal, a value that is good for everybody or for most of the members; for example, the interest may be to avoid accidents. The use of goals of a multiagent system can be explained by the notion of social delegation [77]. *Social delegation* describes the behavior of a social group or institution where some of the agents, on behalf of the other ones, have to achieve some goal which is part of the plans of all members of the group or institution. In this interpretation, the use of sanctions can be explained by the notion of *social control*, "an incessant local (micro) activity of its units" [74], aimed at restoring the regularities prescribed by norms, because a dynamic social order requires a continuous activity for ensuring that the normative system's goals are achieved. In case of sanction-based obligations, this ability is required since the application of sanctions in response to violations cannot be taken for granted.

For example, consider a peer-to-peer system like Napster. Each individual agent does not want that all other agents start to download files from their computer system. From the individual desires that other agents do not all use only their side, we get to a shared or group desire that downloads are evenly distributed over the network. In a normative system, this shared desire may

lead to a norm which says that one should not use computers which already are used intensively. The agents accept this norm, because they rationally see that this increases their own benefits. This example of social delegation is of course analogous to the development of moral principles based on the Kantian imperative: do not do to others what you could not accept yourself. Since, in some cases, for the agents it is rational to violate the norm, sanctions are added to enforce their respect. In our peer-to-peer example, those agents who do not share enough files get a reduced download speed rate.

As a more abstract example involving sanctions, consider the widely discussed prisoner's dilemma. In this example, both agents are better of if they cooperate, but rationality tells them to defect (since this is the only Nash equilibrium). Since there is a shared desire that both agents cooperate, in a normative system a norm will be created that counts defection as a violation that will be sanctioned. The agents will accept this norm and the related sanction, because rationality tells them that they will be better off. In this example, the sanction should be high enough to deter the agents from defecting, for which we can use the game described above.

## 4 Results

This section is structured along the complexity of the interaction between the normative system and the agents, as proposed in [23]. In the simplest setting there is only a single agent and a single normative system it is subject too. Then we consider the role of agents in a normative system (creating or enforcing norms, like legislators, policeman, judges, etc. in legal systems). We then also consider interaction among agents in contracting settings (where the norms of contract live in legal setting of contract frames or legal institutions) and, finally, we consider the most complex case of multiple normative systems (country law vs European law, virtual communities, etc.). Each subsection extends the theory developed in the previous subsection.

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

In [46] we start with considering the normative system from the (subjective) viewpoint of an agent. We also test our model in Tennenholtz artificial social systems as enforceable social laws [43].

*Problems*: Normative systems can be divided in preventative and detective ones. In the former, a system is built such that violations are impossible (you cannot enter metro station without a ticket), in the latter violations can be detected (you can enter train without a ticket but you may be checked and sanctioned). When we accept that norms can be violated, then we go beyond classical cryptographic security into the area of risk analysis, fraud, deception, trust and reputation, and so on. In all these areas, we have to find a way to

asses risk, anticipate behavior of other agents, and so on. Numbers are usually not available, so we develop a qualitative game theory for agents in normative systems.

*Technique*: Introduction of the notion of obligation and its relation with sanctions. Combination of normative reasoning and game theory. Analysis of the way an agent can violate obligations without being sanctioned by the normative system. Analysis of contrary-to-duty reasoning.

*Case-studies.* Transaction trust in normative multiagent systems [19], and energy markets [20].

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

We study permission and authorization in policies for virtual communities of agents [49], and local versus global policies and centralized versus decentralized control in virtual communities of agents [41].

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

In [58] we consider social phenomena in normative multiagent systems like directed and group obligations, how group obligations can be distributed and how norms can emerge. We consider also argument games for interactive access control [21], and negotiation of the distribution of obligations with sanctions among autonomous agents [38,56].

*Application area*: Contracting among agents in e-commerce applications.

*Problem*: In an open multi-agent system agents should be able to constrain the behavior of other agents with whom they are interacting, for example when making economic transactions. Thus it is necessary to have a way to enable agents to create new obligations which are enforced by the normative system. Contracts define the creation of new obligations and permissions by agents involved in a negotiation. The possibility to create new norms is defined by the legal system itself, using constitutive norms.

*Technique*: Introducing constitutive norms and their relation to regulative norms like obligations and permissions. Constitutive or counts-as norms have first been studied in speech act theory, and theories of construction of social reality. They define so-called institutional facts, and we extend their definition so they also define the way in which normative systems can change.

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

We consider the agents from the viewpoint of the normative system: it not only views the agents as subjects which it obliges, prohibits and permits, sanctions

and controls, but it also views the agents as subjects that play roles in the system, for example which can modify the normative system [35]. We consider also game specification in normative multiagent systems based on the Trias Politica [39], and the evolution of artificial Social Systems [44],

*Application area*: Legal systems. Introducing norms in multi-agent systems requires a correct understanding of how norms work in human society. It is necessary to build formal models which match the complexities of legal systems.

*Problems*: Open multi-agent systems cannot be regulated off-line since new unforeseen situations emerge and new norms must be added runtime to deal with specific situations without having to modify the entire set of norms. Legal systems cannot be composed only of obligations, since it becomes too complex to specify new exceptional cases which are not subject to the obligations. So permissions are introduced as a way to specify exceptions. Moreover, the relation between permissions and obligations issued by different levels of hierarchical normative systems is considered.

*Technique*: Introducing permissions and their relation to obligations. Introducing hierarchical normative systems and priority relations among norms. Permissions as exceptions are defined in the input/output logic framework [63], and the notion of authorization is studied [48].

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

In [62] we consider systems that contain multiple normative systems. We consider also the delegation of control to autonomous agents (called defender agents) [33].

*Application area*: Regulating virtual communities of peer agents by means of policies, like in Secure knowledge management.

*Problem*: In scenarios like grid architectures or peer to peer systems every node can be seen as an autonomous agent. As such it can impose freely norms on the use of its resources. When the agent joins a virtual community, it must offer its resources at disposal of the other members according to the global policies of the community. Global policies about local policies, however, have not been studied yet, since they require nesting of norms.

*Technique*: When there are several normative systems, then global norms may put some restrictions on the possibility of issuing local norms. For example, European law limits the way European countries can create and enforce norms. The problem is how to describe global policies on local policies. Our game theoretic analysis illustrates that a naive approach does not work, and we therefore have to formalize an idea suggested by von Wright called delegation of will.

### 4.6 Related topics

Our work on the game-theoretic approach to normative multiagent systems has led us to study some related topics, where we have obtained the following results.

- Model of emergence of norms called the social delegation cycle [64], including a study of the use of power in norm negotiation [66].
- A foundational ontology of organizations and roles [57], an agent oriented ontology of social reality [36], and a model of organizations as socially constructed agents in the agent oriented paradigm [47]. We study the attribution of mental attitudes to roles [37] and we define groups as agents with mental attitudes [40]. We also consider the representation of organizations in artificial social systems [61].
  We develop an extension of the programming language Java called powerJava [10]. We consider power in powerJava [8] and interaction among objects [12]. We bridge agent theory and object orientation by importing social roles in object oriented languages [7] and by agent-like communication among objects [11].
- Based on various social viewpoints on multiagent systems [28] we consider definitions of coalitions [30,31] based on admissible agreements among goal-directed agents [25]. We consider reduction of coalition structures via contracts' representation [29] and an abstraction from power to coalition structures [27].
- We consider the role of roles in agent communication languages [24], and propose role based semantics as a synthesis between mental attitudes and social commitments [15,16,25]. We distinguish propositional and action commitment in agent communication [17]. We consider FIPA communicative acts also in defeasible logic [18].

## 5 Interdisciplinary aspects

Normative multiagent systems are an example of the use of sociological theories in multiagent systems, and more generally of the relation between agent theory and the social sciences such as sociology, philosophy, economics, and legal science. Other examples are the use of the social concept of "power" in programming language powerJava, see also [9,13]. As Castelfranchi [75] argues , not only can social and cognitive concepts like norms and beliefs should be used in agent theory, but agent theory should be used to enrich social sciences, making it more computational. For example, in the area of coordination and organization [53], organizational concepts may be used in coordination languages, or these languages may be used for human organizations. However, our theory is a computer science approach. It may be used outside computer science, but we restrict ourselves to computer science. For an example of how our theory may be used in cognitive science see [45]. For some new challenges for deontic logic due to our theory, see [22,34].

Our game-theoretic approach to normative systems combines deontic logic, agent architecture and artificial social systems. Deontic logic in computer science is an area of knowledge representation and reasoning within artificial intelligence, and related to philosophical logic. Agent architecture is an area of artificial intelligence related to software engineering and psychology. Artificial social systems

are a kind of game theory related to economics, linguistics and organizational theory.

The need for social science theories and concepts like norms in multiagent systems is now well established. For example, Wooldridge's weak notion of agency is based on flexible autonomous action [115], and social ability as the interaction with other agents and co-operation is one of the three meanings of flexibility; the other two are reactivity as interaction with the environment, and pro-activeness as taking the initiative. In this definition autonomy refers to non-social aspects, such as operating without the direct intervention of humans or others, and have some kind of control over their actions and internal state. For some other arguments for the need for social theory in multiagent systems, see, for example, [69,79,114]. For a more complete discussion on the need of social theory in general, and norms in particular, see the AgentLink roadmap [93].

Social concepts like norms are important for multiagent systems, because multiagent system research and sociology share the interest in the relation between micro-level agent behaviour and macro-level system effects. In sociology this is the (in)famous micro-macro link [3] that focuses on the relation between individual agent behaviour and characteristics at the level of the social system. In multiagent system research, this boils down to the question "How to ensure efficiency at the level of the multiagent system whilst respecting individual autonomy?". According to Verhagen [113] three possible solutions to this problem comprise of the use of central control which gravely jeopardizes the agent's autonomy, internalized control like the use of social laws [106], and structural coordination [103] including learning norms.

## 6 Summary

Normative multi-agent systems study general and domain independent properties of norms. It builds on results obtained in deontic logic, the logic of obligations and permissions, for the representation of norms as rules, the application of such rules, contrary-to-duty reasoning and the relation to permissions. However, it goes beyond logical relations among obligations and permissions by explaining the relation among social norms and obligations, relating regulative norms to constitutive norms, explaining the evolution of normative systems, and much more. Our work on the game-theoretic approach to normative multi-agent systems has the following four objectives:

1. How to combine the three existing representations of normative multiagent systems? A representation of a normative multiagent systems that builds on and extends the deontic logic approach for the explicit representation of norms, the agent architecture approach for software engineering of agents, and the game-theoretic artificial social systems approach for model of interaction.
2. Why do we need norms in computer science? A logical framework for qualitative risk analysis, building on ideas in security and risk management.

3. When do we need which kind of norms? A classification of situations in which one needs which elements of normative systems (e.g., obligations, permissions, counts-as conditionals).
4. How can we use norms in computer science applications? Examples and classification of which kind of normative multiagent systems are to be used for which kind of applications in computer science.

In our approach we use input/output logic for deontic logic, BOID architecture for the agent architecture, and both game theory and recursive modeling for artificial social systems. The following table summarizes our results:

| Behavior | violation games | institutional games | negotiation games | norm creation games | control games |
|---|---|---|---|---|---|
| Structure | Interacting with NS | | MAS interaction in NS | MAS structure of NS | Interaction among NSs |
| Mechanism | Obligation | Counts-as | Norm creation actions | Permission | nested rules |
| Theory | BDI, deontic logic | | | | |
| Application | Trust Fraud, deception grid, p2p | Distributed systems Virtual communities | e-commerce e-contracting | security, SKM, policies | |

In our work, we use game theory as basis of NMAS. However, vice versa, it is an open problem whether our model / approach also has something to say about the role of norms in game theory.

## References


1. Alchourrón, C., "Philosophical foundations of deontic logic and the logic of defeasible conditionals", in: Meyer, J.-J. and Wieringa, R. (eds.), *Deontic Logic in Computer Science: Normative System Specification*, John Wiley & Sons, 1993, 43–84.
2. Alchourrón, C. and Bulygin, E., *Normative Systems*, Wien: Springer, 1971.
3. Alexander, J. C., Giesen, B., Munch, R. and Smelser, N. J., *The micro-macro link*, Berkeley: University of California Press, 1987.
4. Anderson, A., "A reduction of deontic logic to Alethic modal logic", *Mind*, **67**, 1958, 100–103.
5. Anderson, A., "Some nasty problems in the formalization of ethics", *Noûs*, **1**, 1967, 345–360.
6. Aqvist, L., "Deontic Logic", in: Gabbay, D. and Guenthner, F. (eds.), *Handbook of Philosophical Logic: Volume II: Extensions of Classical Logic*, Dordrecht: Reidel, 1984, 605–714.
7. Baldoni, M., Boella, G. and van der Torre, L., "Bridging Agent Theory and Object Orientation: Importing Social Roles in Object Oriented Languages", in: *Programming Multi-Agent Systems, Third International Workshop, ProMAS 2005*, vol. 3862 of *LNCS*, Berlin: Springer, 2006, 57–75.
8. Baldoni, M., Boella, G. and van der Torre, L., "I fondamenti ontologici dei linguaggi di programmazione orientati agli oggetti: i casi delle relazioni e dei ruoli", *Networks, rivista di filosofia dell'intelligenza artificiale e scienze cognitive*, **6**, 2006.
9. Baldoni, M., Boella, G. and van der Torre, L., "Modelling the Interaction Between Objects: Roles as Affordances.", in: *Knowledge Science, Engineering and Management, First International Conference, KSEM 2006*, vol. 4092 of *LNCS*, Springer, 2006, 42–54.

10. Baldoni, M., Boella, G. and van der Torre, L., "Roles as a Coordination Construct: Introducing powerJava", *Electronic Notes in Theoretical Computer Science (ENTCS) Procs. of the First International Workshop on Methods and Tools for Coordinating Concurrent, Distributed and Mobile Systems (MTCoord 2005)*, **150**(1), 2006, 9–29.
11. Baldoni, M., Boella, G. and van der Torre, L., "Bridging Agent Theory and Object Orientation: Interaction among Objects", in: *Programming Multi-Agent Systems, Fourth International Workshop, ProMAS 2006*, vol. 4411 of *LNCS*, Springer, 2007, 151–166.
12. Baldoni, M., Boella, G. and van der Torre, L., "Interaction between Objects in powerJava", *Journal of Object Technology*, **6**(2), 2007, 7–12.
13. Baldoni, M., Boella, G. and van der Torre, L., "Relationships meet their roles in object oriented programming", in: *Procs. of the 2nd International Symposium on Fundamentals of Software Engineering 2007 Theory and Practice (FSEN '07)*, 2007.
14. Beccaria, C., *Dei delitti e delle pene*, Livorno, 1764.
15. Boella, G., Damiano, R., Hulstijn, J. and van der Torre, L., "ACL Semantics between Social Commitments and Mental Attitudes", in: *International Workshops on Agent Communication, AC 2005 and AC 2006*, vol. 3859 of *LNAI*, Berlin: Springer, 2006, 30–44.
16. Boella, G., Damiano, R., Hulstijn, J. and van der Torre, L., "Role-Based Semantics for Agent Communication: Embedding of the Mental Attitudes and Social Commitments Semantics", in: *Procs. of the 5th International Joint Conference on Autonomous Agents and Multiagent Systems (AAMAS'06)*, New York (NJ): ACM, 2006, 688–690.
17. Boella, G., Damiano, R., Hulstijn, J. and van der Torre, L., "Distinguishing Propositional and Action Commitment in Agent Communication", in: *Procs. of the 7th Workshop on Computational Models of Natural Argument (CMNA'07)*, 2007.
18. Boella, G., Hulstijn, J., Governatori, G., Riveret, R., Rotolo, A. and van der Torre, L., "FIPA Communicative Acts in Defeasible Logic", in: *Procs. of the 7th International Workshop on Nonmontonic Reasoning, Action and Change (NRAC'07)*, 2007.
19. Boella, G., Hulstijn, J., Tan, Y. and van der Torre, L., "Transaction trust in normative multiagent systems", in: *Procs. of Trust in Agent Societies Workshop at AAMAS'05*, 2005.
20. Boella, G., Hulstijn, J., Tan, Y. and van der Torre, L., "Modeling Control Mechanisms with Normative Multiagent Systems: The Case of the Renewables Obligation", in: *Coordination, Organizations, Institutions, and Norms in Multi-Agent Systems AAMAS 2005 International Workshops on Agents, Norms, and Institutions for Regulated Multiagent Systems, ANIREM 2005 and on Organizations in Multi-Agent Systems, OOOP 2005*, vol. 3913 of *LNAI*, Berlin: Springer, 2006, 114–126.
21. Boella, G., Hulstijn, J. and van der Torre, L., "Argumentation for Access Control.", in: *AI*IA 2005: Advances in Artificial Intelligence, 9th Congress of the Italian Association for Artificial Intelligence*, vol. 3673 of *LNCS*, Berlin: Springer, 2005, 86–97.
22. Boella, G., Hulstijn, J. and van der Torre, L., "Interaction in Normative Multi-Agent Systems.", *Electronic Notes in Theoretical Computer Science, Proceedings of the Workshop on the Foundations of Interactive Computation (FInCo 2005)*, **141**(5), 2005, 135–162.

23. Boella, G., Hulstijn, J. and van der Torre, L., "Virtual organizations as normative multiagent systems", in: *Procs. of the 38th Hawaii International Conference on System Sciences (HICSS-38 2005)*, 2005.
24. Boella, G., Hulstijn, J. and van der Torre, L., "The Roles of Roles in Agent Communication Languages", in: *Procs. of the IEEE/WIC/ACM International Conference on Intelligent Agent Technology (IAT'06)*, IEEE, 2006, 381–384.
25. Boella, G., Hulstjin, J. and van der Torre, L., "A Synthesis Between Mental Attitudes and Social Commitments in Agent Communication Languages", in: *Procs. of the IEEE/WIC/ACM International Conference on Intelligent Agent Technology (IAT'05)*, IEEE, 2005, 358–364.
26. Boella, G. and Lesmo, L., "A game theoretic approach to norms", *Cognitive Science Quarterly*, **2(3-4)**, 2002, 492–512.
27. Boella, G., Sauro, L. and van der Torre, L., "An Abstraction from Power to Coalition Structures", in: *Procs. of the 16th European Conference on Artificial Intelligence (ECAI'04)*, Amsterdam: IOS, 2004, 965–966.
28. Boella, G., Sauro, L. and van der Torre, L., "Social Viewpoints on Multiagent Systems", in: *Procs. of the 3rd International Joint Conference on Autonomous Agents and Multiagent Systems (AAMAS'04)*, New York (NJ): ACM, 2004, 1358–1359.
29. Boella, G., Sauro, L. and van der Torre, L., "Reducing Coalition Structures via contracts' representation", in: *Procs. of the 4th International Joint Conference on Autonomous Agents and Multiagent Systems (AAMAS'05)*, New York (NJ): ACM, 2005, 1187–1188.
30. Boella, G., Sauro, L. and van der Torre, L., "Strengthening Admissible Coalitions", in: *Procs. of the 17th European Conference on Artificial Intelligence (ECAI'06)*, Amsterdam: IOS, 2006, 195–199.
31. Boella, G., Sauro, L. and van der Torre, L. W. N., "From Social Power to Social Importance", *Web Intelligence and Agent Systems Journal (WIAS)*, 2007.
32. Boella, G. and van der Torre, L., "Attributing mental attitudes to normative systems", in: *Procs. of the 2nd International Joint Conference on Autonomous Agents and Multiagent Systems (AAMAS'03)*, New York (NJ): ACM Press, 2003, 942–943.
33. Boella, G. and van der Torre, L., "Norm Governed Multiagent Systems: The delegation of control to autonomous agents", in: *Proceedings of the 2003 IEEE/WIC International Conference on Intelligent Agent Technology (IAT'03)*, IEEE, 2003, 329– 335.
34. Boella, G. and van der Torre, L., "Obligations as Social Constructs", in: *AI*IA 2003 - Advances in Artificial Intelligence, 8th Congress of the Italian Association for Artificial Intelligence*, vol. 2829 of *LNAI*, Berlin: Springer, 2003, 27–38.
35. Boella, G. and van der Torre, L., "Rational norm creation: Attributing mental attitudes to normative systems, part 2", in: *Procs. of the 8th International Conference on Artificial Intelligence and Law (ICAIL'03)*, New York (NJ): ACM Press, 2003, 81–82.
36. Boella, G. and van der Torre, L., "An agent oriented ontology of social reality", in: *Procs. of Formal Ontologies in Information Systems (FOIS'04)*, Amsterdam: IOS, 2004, 199–209.
37. Boella, G. and van der Torre, L., "Attributing Mental Attitudes to Roles: The Agent Metaphor Applied to Organizational Design", in: *Procs. of the 6th International Conference on Electronic Commerce (ICEC'04)*, New York (NJ): ACM, 2004, 130–137.

38. Boella, G. and van der Torre, L., "The Distribution of Obligations by Negotiation among Autonomous Agents", in: *Procs. of the 16th European Conference on Artificial Intelligence (ECAI'04)*, Amsterdam: IOS, 2004, 13–17.
39. Boella, G. and van der Torre, L., "Game Specification in Normative Multiagent System: the Trias Politica", in: *Procs. of IEEE/WIC/ACM International Conference on Intelligent Agent Technology (IAT'04)*, IEEE, 2004, 504–508.
40. Boella, G. and van der Torre, L., "Groups as Agents with Mental Attitudes", in: *Procs. of the 3rd International Joint Conference on Autonomous Agents and Multiagent Systems (AAMAS'04)*, New York (NJ): ACM, 2004, 964–971.
41. Boella, G. and van der Torre, L., "Local vs Global Policies and Centralized vs Decentralized Control in Virtual Communities of Agents", in: *Procs. of IEEE/WIC/ACM International Conference on Web Intelligence (WI'04)*, IEEE, 2004, 690–693.
42. Boella, G. and van der Torre, L., "Regulative and Constitutive Norms in Normative Multiagent Systems", in: *Procs. of the 10th International Conference on the Principles of Knowledge Representation and Reasoning KR'04*, Menlo Park (CA): AAAI, 2004, 255–265.
43. Boella, G. and van der Torre, L., "Enforceable social laws", in: *Procs. of 4th International Joint Conference on Autonomous Agents and Multiagent Systems (AAMAS'05)*, New York (NJ): ACM, 2005, 682–689.
44. Boella, G. and van der Torre, L., "The Evolution of Artificial Social Systems", in: *Procs. of the 19th International Joint Conference on Artificial Intelligence (IJCAI'05)*, Professional Book Center, 2005, 1655–1556.
45. Boella, G. and van der Torre, L., "From the Theory of Mind to the Construction of Social Reality", in: *Procs. of the 27th Annual Conference of the Cognitive Science Society (CogSci'05)*, Mahwah (NJ): Lawrence Erlbaum, 2005, 298–303.
46. Boella, G. and van der Torre, L., "Normative multiagent systems and trust dynamics", in: *Trusting Agents for Trusting Electronic Societies, Theory and Applications in HCI and E-Commerce*, vol. 3577 of *LNAI*, Berlin: Springer, 2005, 1–17.
47. Boella, G. and van der Torre, L., "Organizations as Socially Constructed Agents in the Agent Oriented Paradigm", in: *Engineering Societies in the Agents World V, 5th International Workshop (ESAW'04)*, vol. 3451 of *LNAI*, Berlin: Springer, 2005, 1–13.
48. Boella, G. and van der Torre, L., "Permission and Authorization in Normative Multiagent Systems", in: *Procs. of the 10th International Conference on Artificial Intelligence and Law (ICAIL'05)*, New York (NJ): ACM, 2005, 236–237.
49. Boella, G. and van der Torre, L., "Permission and Authorization in Policies for Virtual Communities of Agents", in: *Agents and Peer-to-Peer Computing, Third International Workshop (AP2PC'04)*, vol. 3601 of *LNCS*, Berlin: Springer, 2005, 86–97.
50. Boella, G. and van der Torre, L., "Role-based rights in artificial social systems", in: *Procs. of the IEEE/WIC/ACM International Conference on Intelligent Agent Technology (IAT'05)*, IEEE, 2005, 516–519.
51. Boella, G. and van der Torre, L., "An architecture of a normative system", in: *Procs. of the 5th International Joint Conference on Autonomous Agents and Multiagent Systems (AAMAS'06)*, New York (NJ): ACM, 2006, 229–231.
52. Boella, G. and van der Torre, L., "Constitutive Norms in the Design of Normative Multiagent Systems", in: *Computational Logic in Multi-Agent Systems, 6th International Workshop (CLIMA VI)*, vol. 3900 of *LNCS*, Berlin: Springer, 2006, 303–319.

53. Boella, G. and van der Torre, L., "Coordination and Organization: Definitions, Examples and Future Research Directions.", *Electronic Notes in Theoretical Computer Science (ENTCS) Procs. of the First International Workshop on Coordination and Organisation (CoOrg 2005)*, **150**(3), 2006, 3–20.
54. Boella, G. and van der Torre, L., "Count-As Conditionals, Classification and Context.", in: *Procs. of the 17th European Conference on Artificial Intelligence (ECAI'06)*, Amsterdam: IOS, 2006, 719–720.
55. Boella, G. and van der Torre, L., "Delegation of Power in Normative Multiagent Systems", in: *Deontic Logic and Artificial Normative Systems, 8th International Workshop on Deontic Logic in Computer Science (ΔEON'06)*, vol. 4048 of *LNCS*, Berlin: Springer, 2006, 36–52.
56. Boella, G. and van der Torre, L., "Fair Distribution of Collective Obligations.", in: *Procs. of the 17th European Conference on Artificial Intelligence(ECAI'06)*, Amsterdam: IOS, 2006, 721–722.
57. Boella, G. and van der Torre, L., "A Foundational Ontology of Organizations and Roles", in: *Declarative Agent Languages and Technologies IV, 4th International Workshop (DALT'06)*, vol. 4327 of *LNCS*, 2006, 78–88.
58. Boella, G. and van der Torre, L., "A Game Theoretic Approach to Contracts in Multiagent Systems", *IEEE Transactions on Systems, Man and Cybernetics - Part C: Applications and Reviews*, **36**(1), 2006, 68–79.
59. Boella, G. and van der Torre, L., "Game-Theoretic Foundations for Norms", in: *Procs. of Artificial Intelligence Studies*, vol. 3(26), 2006, 39–51.
60. Boella, G. and van der Torre, L., "A Logical Architecture of a Normative System", in: *Deontic Logic and Artificial Normative Systems, 8th International Workshop on Deontic Logic in Computer Science (ΔEON'06)*, vol. 4048 of *LNCS*, Berlin: Springer, 2006, 24–35.
61. Boella, G. and van der Torre, L., "Organizations in Artificial Social Systems", in: *Coordination, Organizations, Institutions, and Norms in Multi-Agent Systems AAMAS 2005 International Workshops on Agents, Norms, and Institutions for Regulated Multiagent Systems, ANIREM 2005 and on Organizations in Multi-Agent Systems, OOOP 2005*, vol. 3913 of *LNAI*, Berlin: Springer, 2006, 198–210.
62. Boella, G. and van der Torre, L., "Security Policies for Sharing Knowledge in Virtual Communities", *IEEE Transactions on Systems, Man and Cybernetics - Part A: Systems and Humans*, **36**(3), 2006, 439–450.
63. Boella, G. and van der Torre, L., "Institutions with a Hierarchy of Authorities in Distributed Dynamic Environments", *Artificial Intelligence and Law Journal (AILaw)*, 2007.
64. Boella, G. and van der Torre, L., "Norm Negotiation in Multiagent Systems", *International Journal of cooperative Information Systems (IJCIS) Special Issue: Emergent Agent Societies*, **16**(2), 2007, 97–122.
65. Boella, G. and van der Torre, L., "The Ontological Properties of Social Roles in Multi-agent Systems: Definitional Dependence, Powers and Roles Playing Roles", *Artificial Intelligence and Law Journal (AILaw)*, 2007.
66. Boella, G. and van der Torre, L., "Power in Norm Negotiation", in: *Procs. of the 1st KES Symposium on Agent and Multi-Agent Systems Technologies and Applications (KES-AMSTA'07)*, LNCS, Berlin: Springer, 2007.
67. Boella, G. and van der Torre, L., "Substantive and Procedural Norms in Normative Multiagent Systems", *Journal of Applied Logic*, 2008.
68. Boella, G., van der Torre, L. and Verhagen, H., "Introduction to normative multiagent systems", *Computation and Mathematical Organizational Theory, Special issue on Normative Multiagent Systems*, **12**(2-3), 2006, 71–79.

69. Bond, A. and Gasser, L., "An Analysis of Problems and Research in DAI", in: *Readings in Distributed Artificial Intelligence*, San Mateo (CA): Morgan Kaufmann, 1988, 3–35.
70. Brafman, R. and Tennenholtz, M., "On Partially Controlled Multi-Agent Systems.", *Journal of Artificial Intelligence Research (JAIR)*, **4**, 1996, 477–507.
71. Bratman, M., *Intentions, plans, and practical reason*, Harvard (Massachusetts): Harvard University Press, 1987.
72. Broersen, J., Dastani, M., Hulstijn, J. and van der Torre, L., "Goal generation in the BOID architecture", *Cognitive Science Quarterly*, **2(3-4)**, 2002, 428–447.
73. Broersen, J., Dastani, M., Hulstijn, J. and van der Torre, L., "Goal generation in the BOID architecture", *Cognitive Science Quarterly*, **2(3-4)**, 2002, 428–447.
74. Castelfranchi, C., "Engineering social order", in: *Engineering Societies in the Agent World, First International Workshop (ESAW'00)*, vol. 1972 of *LNAI*, Berlin: Springer, 2000, 1–18.
75. Castelfranchi, C., "The micro-macro constitution of power", *Protosociology*, **18**, 2003, 208–269.
76. Conte, R., Castelfranchi, C. and Dignum, F., "Autonomous norm-acceptance", in: *Intelligent Agents V (ATAL'98)*, vol. 1555 of *LNCS*, Berlin: Springer, 1999, 99–112.
77. Ferber, J., Gutknecht, O., Jonker, C., Müller, J. P. and Treur, J., "Organization Models and Behavioural Requirements Specification for Multi-Agent Systems", in: *Procs. of Modelling Autonomous Agents in a Multi-Agent World - 10th European Workshop on Multi-Agent Systems (MAAMAW'01)*, 2002.
78. Ferber, J., Gutknecht, O. and Michel, F., "From Agents to Organizations: an organizational view of multiagent systems", in: *Agent-Oriented Software Engineering IV, 4th International Workshop (AOSE'03)*, vol. 2935 of *LNCS*, Berlin: Springer, 2003, 214–230.
79. Gilbert, N. and Conte, R., *Artificial Societies: The Computer Simulation of Social Life*, London: UCL Press, 1995.
80. Gmytrasiewicz, P. J. and Durfee, E. H., "Formalization of recursive modeling", in: *Procs. of the 1st International Conference on Multiagent Systems (ICMAS'95)*, Cambridge (MA): AAAI/MIT Press, 1995, 125–132.
81. Gneezy, U. and Rustichini, A., "A Fine is a Price", *The Journal of Legal Studies*, **29**(1), 2000, 1–18.
82. Goffman, E., *Strategic Interaction*, Oxford: Basil Blackwell, 1970.
83. Gordijn, J. and Tan, Y.-H., "A Design Methodology for Trust and Value Exchanges in Business Models", in: *Procs. of BLED Conference*, 2003, 423–432.
84. Grossi, D., Dignum, F. and Meyer, J., "Contextual Terminologies", in: *Computational Logic in Multi-Agent Systems, 6th International Workshop (CLIMA VI)*, vol. 3900 of *LNCS*, Berlin: Springer, 2006, 284–302.
85. Grossi, D., Meyer, J.-J. and Dignum, F., "Counts-as: Classification or Constitution? An Answer Using Modal Logic", in: *Deontic Logic and Artificial Normative Systems, 8th International Workshop on Deontic Logic in Computer Science (ΔEON'06)*, vol. 4048 of *LNCS*, Berlin: Springer, 2006, 115–130.
86. Hart, H., *The Concept of Law*, Oxford: Clarendon Press, 1961.
87. Hollis, M., *Trust within reason*, Cambridge: Cambridge University Press, 1998.
88. Jennings, N. R., "On Agent-Based Software Engineering", *Artificial Intelligence*, **117(2)**, 2000, 277–296.
89. Jones, A. and Carmo, J., "Deontic logic and Contrary-to-Duties", in: Gabbay, D. and Guenthner, F. (eds.), *Handbook of Philosophical Logic*, vol. 3, Dordrecht (NL): Kluwer, 2001, 203–279.

90. Jones, A. and Sergot, M., "A Formal Characterisation of Institutionalised Power", *Journal of IGPL*, **3**, 1996, 427–443.
91. Lee, R., "Documentary Petri Nets: A Modeling Representation for Electronic Trade Procedures", in: *Business Process Management, Models, Techniques, and Empirical Studies*, vol. 1806 of *LNCS*, Berlin: Springer, 2000, 359–375.
92. Levitt, S. D. and Dubner, S. J., *Freakonomics : A Rogue Economist Explores the Hidden Side of Everything*, New York: William Morrow, 2005.
93. Luck, M., McBurney, P. and Preist, C., *Agent Technology: Enabling Next Generation Computing (A Roadmap for Agent Based Computing)*, AgentLink, 2003.
94. Makinson, D. and van der Torre, L., "Input-output logics", *Journal of Philosophical Logic*, **29**(4), 2000, 383–408.
95. Makinson, D. and van der Torre, L., "Constraints for input-output logics", *Journal of Philosophical Logic*, **30(2)**, 2001, 155–185.
96. Makinson, D. and van der Torre, L., "Permissions from an input-output perspective", *Journal of Philosophical Logic*, **32(4)**, 2003, 391–416.
97. McCallum, M., Norman, T. and Vasconcelos, W., "A Formal Model of Organisations for Engineering Multi-Agent Systems", in: *Procs. of Coordination in Emergent Agent Societies Workshop (CEAS'04)*, 2004.
98. Merriam-Webster, *Dictionary of Law*, Merriam-Webster, 1996.
99. Merriam-Webster, *On Line Dictionary*, Merriam-Webster, 2007.
100. Meyer, J.-J. and Wieringa, R., *Deontic Logic in Computer Science: Normative System Specification*, Chichester, England: John Wiley & Sons, 1993.
101. Meyer, J. J. C., "A Different Approach to Deontic Logic: Deontic Logic Viewed as a Variant of Dynamic Logic", *Notre Dame Journal of Formal Logic*, **29(1)**, 1988, 109–136.
102. Newell, A., "The knowledge level", *Artificial Intelligence*, **18**, 1982, 87–127.
103. Ossowski, S., *Co-Ordination in Artificial Agent Societies: Social Structures and Its Implications for Autonomous Problem-Solving Agents*, Berlin: Springer, 1999.
104. Ruiter, D., "A basic classification of legal institutions", *Ratio Juris*, **10(4)**, 1997, 357–371.
105. Searle, J., *The Construction of Social Reality*, New York: The Free Press, 1995.
106. Shoham, Y. and Tennenholtz, M., "On Social Laws for Artificial Agent Societies: Off-Line Design", *Artificial Intelligence*, **73**(1-2), 1995, 231–252.
107. Shoham, Y. and Tennenholtz, M., "On the Emergence of Social Conventions: Modeling, Analysis and Simulations", *Artificial Intelligence*, **94**(1–2), 1997, 139–166.
108. Tennenholtz, M., "On Stable Social Laws and Qualitative Equilibria", *Artificial Intelligence*, **102(1)**, 1998, 1–20.
109. van der Torre, L., "Contextual Deontic Logic: Normative Agents, Violations and Independence", *Annals of Mathematics and Artificial Intelligence*, **37(1-2)**, 2003, 33–63.
110. van der Torre, L. and Tan, Y., "Contrary-To-Duty Reasoning with Preference-based Dyadic Obligations", *Annals of Mathematics and Artificial Intelligence*, **27**(1-4), 1999, 49–78.
111. Tuomela, R., *Cooperation: A Philosophical Study*, Dordrecht: Kluwer, 2000.
112. Verhagen, H., "On the learning of norms", in: *Procs. of MultiAgent System Engineering, 9th European Workshop on Modelling Autonomous Agents in a Multi-Agent World (MAAMAW'99)*, vol. 1647 of *LNCS*, Berlin: Springer, 1999.
113. Verhagen, H., *Norm Autonomous Agents*, Ph.D. thesis, Stockholm University, 2000.

114. Verhagen, H. and Smit, R., "Multiagent systems as simulation tools for social theory testing", in: *Procs. of International Conference on Computer Simulation and the Social Sciences (ISSC&SS'97)*, 1997.
115. Wooldridge, M. J. and Jennings, N. R., "Intelligent Agents: Theory and Practice", *Knowledge Engineering Review*, **10**(2), 1995, 115–152.
116. von Wright, G., "Deontic logic - as I see it", in: McNamara, P. and Prakken, H. (eds.), *Norms, Logics and Information Systems. New Studies on Deontic Logic and Computer Science*, IOS, 1999, 15–25.
117. von Wright, G. H., "Deontic logic", *Mind*, **60**, 1951, 1–15.
118. von Wright, G. H., *An Essay in Modal Logic*, Amsterdam: North-Holland, 1951.
119. Zambonelli, F., Jennings, N. and Wooldridge, M., "Developing Multiagent Systems: The Gaia Methodology", *IEEE Transactions of Software Engineering and Methodology*, **12(3)**, 2003, 317–370.

# A Normative Framework for Agent-Based Systems

Fabiola López y López (fabiola.lopez@siu.buap.mx)
*Benemérita Universidad Autónoma de Puebla, México*

Michael Luck (mml@ecs.soton.ac.uk)
*University of Southampton, United Kingdom*

Mark d'Inverno (dinverm@westminster.ac.uk)
*University of Westminster, United Kingdom*

**Abstract.** One of the key issues in the computational representation of *open societies* relates to the introduction of *norms* that help to cope with the heterogeneity, the autonomy and the diversity of interests among their members. Research regarding this issue presents two omissions. One is the lack of a canonical model of norms that facilitates their implementation, and that allows us to describe the processes of reasoning about norms. The other refers to considering, in the model of normative multi-agent systems, the perspective of individual agents and what they might need to effectively reason about the society in which they participate. Both are the concerns of this paper, and the main objective is to present a formal normative framework for agent-based systems that facilitates their implementation.

**Keywords:** normative agents, normative multi-agent systems

## 1. Introduction

Norms have long been used as mechanisms to limit human autonomy in such a way that coexistence between self-interested and untrusted people has been made possible. They are indispensable to overcome problems of coordination of large, complex and heterogeneous systems where total and direct social control cannot be exerted. From this experience, the introduction of *norms* that help to cope with the heterogeneity, the autonomy and the diversity of interests among autonomous agents has been considered as a key issue towards the computational representation of *open societies* of agents (Luck et al., 2003).

The introduction of norms in agents and multi-agent systems is far from trivial. Research on norms and agents has ranged from fundamental work on the importance of norms in agent behaviour (Conte et al., 1999b), to proposing internal representations of norms (Conte and Castelfranchi, 1995), analysing the different types of norms (Dignum, 1999; Tuomela and Bonnevier-Toumela, 1995), considering their emergence in groups of agents (Walker and Wooldridge, 1995), proposing logics for their formalisation (van der Torre

* The first author acknowledges funding from the Faculty Enhancement Program (PROMEP) of the Mexican Ministry of Public Education (SEP), project No PROMEP/103.5/04/767.

Dagstuhl Seminar Proceedings 07122
Normative Multi-agent Systems
http://drops.dagstuhl.de/opus/volltexte/2007/933

and Tan, 1999a; Sergot, 1999; Wieringa et al., 1996), and both analysing and representing institutions controlled by norms (Balzer and Tuomela, 2001; Shoham and Tennenholtz, 1995). Norms can also be analysed from the internal point of view of agents and the role agents play in their processing. In this case, we can describe for example, how agents manage norm adoption and compliance (Boella and Lesmo, 2001; Dignum et al., 2000; López y López et al., 2002), how agents responsible for enforcing norms must behave (Castelfranchi et al., 1998), and what the characteristics are of agents entitled to exert power in a society (Jones and Sergot, 1996). Taking into account the different perspectives from which research has been done, it is no rare to find many disparities and gaps between theories and, consequently, the work for designers is hard when they try to develop applications based on normative multi-agent systems.

For instance, although efforts have been made to describe and define the different types of norms that agents have to deal with (Dignum, 1999; Singh, 1999), work has not led into a model that facilitates the computational representation of any kind of norm. Each kind of norm appears to be different, which also suggests that different processes of reasoning for an agent should be proposed. There are also some work that introduces norms in systems of agents to represent societies, institutions and organisations (Dellarocas and Klein, 2001; Dignum and Dignum, 2001; Esteva et al., 2001; Shoham and Tennenholtz, 1995). This research is primarily focused at the level of multi-agent systems, where norms represent the means to achieve coordination among their members. There, agents are assumed to be able to comply with norms, to adopt new norms, and to obey the authorities of the system but nothing is said about the reasons why agents might be willing to adopt and comply with norms, nor about how agents can identify situations in which an authority's orders are beyond its responsibilities. That is, although agents in such systems are said to be autonomous, their models of norms and systems regulated by norms do not offer the means to explain why *autonomous* agents that are working to satisfy their own goals, still comply with their social responsibilities. The game-theoretical approach of norms (Axelrod, 1986; Bicchieri, 1990; Hashimoto and Egashira, 2001; Ullmann-Margalit, 1977; Walker and Wooldridge, 1995) explains this situation as a result of agents applying strategies that enable them to converge to situations that are beneficial for a group of interacting agents. Having found such a strategy, it becomes a norm for all the members in the group, and since this norm is agreed by all agents, it is always complied with by them. Although interesting, this approach is not useful for the aims of our work, because rather than being concerned with the process of how a specific norm is created, our research focuses on the modelling of different types of norms and on specifying each one of the internal processes that explain the normative behaviour of agents in situations such as the observed by game-theoretical approaches.

Now, although the importance of modelling compliance with norms as an autonomous decision has been identified by several researchers (Castelfranchi et al., 2000; Conte et al., 1999a; Conte and Dellarocas, 2001; Conte et al., 1999b), the issue is only partly addressed by others whose proposals for norm compliance generally rely on specific decision-making strategies based on how much an agent gains or loses by complying with (Barbuceanu et al., 1999; Dignum et al., 2000), and on the probability of being caught by a defender of a norm (Boella and Lesmo, 2001). We consider these cases as very specific and, therefore, inadequate to model different kinds of normative behaviour of autonomous agents.

As a way to overcome these omissions, we have developed a normative framework for agent-based systems that includes a canonical model of norms, a model of normative multi-agent systems and a model of normative autonomous agents. The framework provides the means to understand the normative behaviour of autonomous agents and to facilitate their implementation. Independent components of this framework have already been presented in different forums (López y López and Luck, 2003; López y López and Luck, 2004; López y López et al., 2002; López y López et al., 2004), here the framework is presented as a whole. Moreover, although we recognise the importance of deontic logic as a formalism to represent knowledge and reasoning about the behaviour of agents into systems regulated by norms, we use the Z language, which is based on set-theory and first order logic (Spivey, 1992), to present the formal model of our framework because there are tools that allow verification and methodologies that guide the translation of formalizations into code.

The organisation of this paper is as follows. First, a formal definition of autonomous agents is given. After that, an analysis of different properties of norms is provided. This analysis is then used to justify the elements that a general model of a norm must include in order to enable autonomous agents to reason about them. Next, the main properties of systems of autonomous agents that are regulated by norms are discussed and a model is presented. Then, we describe our proposal to enable agents to reason about norms. Finally, our conclusions are provided.

## 2. Autonomous Agents

The foundations of this work are taken from Luck and d'Inverno's SMART agent framework (d'Inverno and Luck, 2003) whose concept of *motivations* as the driving force that affects the reasoning of agents in satisfying their goals is considered as the underlying argument for agents to voluntarily comply with norms and to voluntarily enter and remain in a society. In the SMART agent framework, an *attribute* represents a perceivable feature of the agent's environment, which can be represented as a predicate or its negation. Then, a particular *state* in the environment is described by a set of attributes, a

*goal* represents situations that an agent wishes to bring about, *motivations* are desires or preferences that affect the outcome of the reasoning intended to satisfy an agent's goals, and *actions* are discrete events that change the state of the environment when performed. For the purposes of this paper, we formally describe environmental states, goals, actions and autonomous agents. Details of the remaining elements are not needed, so we simply consider them as given sets.

$$[Attribute, Motivation]$$

$$EnvState == \mathbb{P}_1\, Attribute$$
$$Goal == \mathbb{P}_1\, Attribute$$
$$Action == EnvState \rightarrow EnvState$$

In the schema below, an autonomous agent is described by a set of goals that it wants to bring about, a set of capabilities that it is able to perform, a non-empty set of motivations representing its preferences, and a set of beliefs representing its vision about the external world. We also assume that the agent is able to determine the $importance$ of its goals, which depends on its current motivations. In this way, the more motivated a goal the higher its importance.

*AutonomousAgent*

$$goals : \mathbb{P}\, Goal;$$
$$capabilities : \mathbb{P}\, Action;$$
$$motivations : \mathbb{P}\, Motivation;$$
$$beliefs : \mathbb{P}_1\, Attribute$$
$$importance : \mathbb{P}(\mathbb{P}\, Goal \times \mathbb{P}\, Motivation) \rightarrow \mathbb{N}$$

---

$$goals \neq \varnothing; \qquad motivations \neq \varnothing$$
$$\forall\, x : \mathbb{P}\, Goal, y : \mathbb{P}\, Motivation \bullet$$
$$(x, y) \in \mathrm{dom}\, importance \mid$$
$$x \subseteq goals \land y \subseteq motivations$$

## 3. Norms

Norms not only have been studied from the philosophical, social and legal points of views (Ross, 1968; Tuomela, 1995) but also from the game-theoretical approach (Axelrod, 1986; Bicchieri, 1990; Hashimoto and Egashira, 2001; Ullmann-Margalit, 1977; Walker and Wooldridge, 1995). Research on norms is also a main issue for both the Artificial Intelligence and the Autonomous Agents communities (Boella and Lesmo, 2001; Castelfranchi et al., 2000; Conte and Castelfranchi, 1995; Dignum et al., 2000; Dignum, 1999; Jones and Sergot, 1996; López y López, 2003; Sergot, 1999; Singh, 1999; Shoham and Tennenholtz, 1995; van der Torre and Tan, 1999a; van der Torre and Tan, 1999b). So, reaching a common definition of norm seems to be

impossible. Nevertheless, we can provide some characteristics for a norm that allow us to identify the main components of a norm and, consequently, to define and represent normative concepts.

Norms facilitate mechanisms to drive the behaviour of agents, especially in those cases when their behaviour affects other agents. Norms can be characterised by their *prescriptiveness*, *sociality*, and *social pressure*. In other words,

- a norm tells an agent how to behave (*prescriptiveness*);
- in situations where more than one agent is involved (*sociality*);
- and since it is always expected that norms conflict with the personal interest of some agents, socially acceptable mechanisms to force agents to comply with norms are needed (*social pressure*).

By analysing these properties, the essential components of a norm can be identified.

### 3.1. Norm Components

Norms specify patterns of behaviour for a set of agents. These patterns are sometimes represented as actions to be performed (Axelrod, 1986; Tuomela, 1995), or restrictions to be imposed over an agent's actions (Norman et al., 1998; Shoham and Tennenholtz, 1995). At other times, patterns of behaviour are specified through goals that must either be satisfied or avoided by agents (Conte and Castelfranchi, 1995; Singh, 1999). Now, since actions are performed in order to change the state of an environment, goals are states that agents want to bring about, and restrictions can be seen as goals to be avoided, we argue that by considering goals the other two patterns of behaviour can be easily represented (as shown in (López y López and Luck, 2003)).

In brief, norms specify things that ought to be done and, consequently, a set of *normative goals* must be included. Sometimes, these normative goals must be directly intended, while at other times their role is to inhibit specific states (as in the case of prohibitions). Norms are always directed at a set of *addressee agents*, which are directly responsible for the satisfaction of the normative goals. Moreover, sometimes to take decisions regarding norms, agents not only consider what must be done but also for whom it must be done. Then, agents that *benefit* from the satisfaction of normative goals may also be included.

In general, norms are not applied all the time, but only in particular circumstances or within a specific *context*. Thus, norms must always specify the situations in which addressee agents must fulfill them. *Exception* states may also be included to represent situations in which addressees cannot be punished when they *have not* complied with norms. Exceptions represent *immunity* states for all addressee agents in a particular situation (Ross, 1968).

Now, to ensure that personal interests do not impede the fulfillment of norms, mechanisms either to promote compliance with norms, or to inhibit deviation from them, are needed. Norms may include *rewards* to be given when normative goals become satisfied, or *punishments* to be applied when they are not. Both rewards and punishments are the means for addressee agents to determine what might happen whatever decision they take regarding norms. They are not the responsibility of addressees agents but of other agents already entitled to either reward or punish compliance and non-compliance with norms. Since rewards and punishments represent states to be achieved, it is natural to consider them as goals but, in contrast with normative goals that must be satisfied by addressees, punishments and rewards are satisfied by agents entitled to do so.

In other words, a norm must be considered for fulfillment by an agent when certain environmental states, not included as exception states, hold. Such a norm forces a group of addressee agents to satisfy some normative goals for a (possibly empty) set of beneficiary agents. In addition, agents are aware that rewards may be enjoyed if norms become satisfied, or that punishments that affect their current goals can be applied if not.

$$
\begin{array}{|l}
\hline
Norm \\
normativegoals : \mathbb{P}\, Goal \\
addressees : \mathbb{P}\, NormativeAgent \\
beneficiaries : \mathbb{P}\, NormativeAgent \\
context : EnvState \\
exceptions : EnvState \\
rewards : \mathbb{P}\, Goal \\
punishments : \mathbb{P}\, Goal \\
\hline
normativegoals \neq \varnothing \\
addressees \neq \varnothing \\
context \neq \varnothing \\
context \cap exceptions = \varnothing \\
rewards \cap punishments = \varnothing \\
\hline
\end{array}
$$

The formal specification of a norm is given in the $Norm$ schema where all the components of norms described above are included, together with some constraints on them. First, it does not make any sense to have norms specifying nothing, norms directed at nobody, or norms that either never or always become applied. Thus, the first three predicates in the schema state that the set of normative goals, the set of addressee agents, and the context must never be empty. The fourth predicate states that the set of attributes describing both the context and exceptions must be disjoint to avoid inconsistencies in identifying whether a norm must be applied. The final constraint

specifies that punishments and rewards are also consistent and, therefore, they must be disjoint.

### 3.2. CONSIDERATIONS

The term *norm* has been used as a synonym for obligations (Boella and Lesmo, 2001; Dignum et al., 2000), prohibitions (Dignum, 1999), social laws (Shoham and Tennenholtz, 1995), and other kinds of rules imposed by societies (or by an authority). The position of our work is quite different. It considers that all these terms can be grouped in a general definition of a norm, because they have the same properties (i.e. prescriptiveness, sociality and social pressure) and they can be represented by the same model. They all represent responsibilities for addressee agents, and create expectations for beneficiaries and other agents. They are also the means to support beneficiaries when they have to claim some compensation in the situations where norms are not fulfilled as expected. Moreover, whatever the kind of norm being considered, its fulfillment may be rewarded, and its violation may be penalised. What makes one norm different from another is the way in which they are created, their persistence, and the components that are obligatory in the norm. Thus, norms might be created by an agent designer as built-in norms, they can be the result of agreements between agents, or they can be elaborated by a complex legal system. Regarding their persistence, norms might be taken into account during different periods of time, such as until an agent dies, as long as an agent stays in a society, or just for a short period of time until its normative goals become satisfied. Finally, some components of a norm might not exist; there are norms that include neither punishments nor rewards, even though they are complied with. Some of these characteristics can be used to provide a *classification* of norms into four main categories: *obligations*, *prohibitions*, *social commitments* and *social codes* (López y López and Luck, 2003). Despite these differences, all types of norms can be reasoned about in similar ways.

Now, to understand the consequences of norms in a particular system, it is necessary to consider norms that are either *fulfilled* or *unfulfilled*. However, since most of the time a norm has a set of agents as addressees, the meaning of fulfilling a norm might depend on the interpretation of analysers of a system. In small groups of agents, it might be easy to consider a norm as fulfilled when every addressee agent has fulfilled the norm; by contrast, in larger societies, a proportion of agents complying with a norm will be enough to consider it as fulfilled. Instead of defining fulfilled norms in general, it is more appropriate to define norms being fulfilled by a particular addressee agent. To do so, the concept of norm instances is introduced as follows. Once a norm is adopted by an agent, a *norm instance* is created, which represents the internalisation of a norm by an agent (Conte and Castelfranchi, 1995). A norm instance is a copy of the original norm that is now used as a *mental*

*attitude* from which new goals for the agent might be inferred. Norms and norm instances are the same concept used for different purposes. Norms are abstract specifications that exist in a society and are known by all agents (Tuomela, 1995), but agents work with *instances* of these norms. Consequently, there must be a separate instance for each addressee of a norm. Due to space constraints, formal definitions and examples of categories of norms, norm instances and fulfilled norms are not provided here but can be found elsewhere (López y López and Luck, 2003).

### 3.3. INTERLOCKING NORMS

The norms of a system are not isolated from each other; sometimes, compliance with them is a condition to trigger (or activate) other norms. That is, there are norms that prescribe how some agents must behave in situations in which other agents either comply with a norm or do not comply with it (Ross, 1968). For example, when employees comply with their obligations in an office, paying their salary becomes an obligation of the employer; or when a plane cannot take-off, providing accommodation to passengers becomes a responsibility of the airline. Norms related in this way can make a complete chain of norms because the newly activated norms can, in turn, activate new ones. Now, since triggering a norm depends on past compliance with another norm, we call these kinds of norms *interlocking norms*. The norm that gives rise to another norm is called the *primary* norm, whereas the norm activated as a result of either the fulfillment or violation of the first is called the *secondary* norm.

In terms of the norm model mentioned earlier, the *context* is a state that must hold for a norm to be complied with. Since the fulfillment of a norm is assessed through its normative goals, the context of the secondary norm must include the satisfaction (or non-satisfaction) of all the primary norm's normative goals. Figure 1 illustrates the structure of both the primary and the secondary norms and how they are interlocked through the primary norm's normative goals and the secondary norm's context.

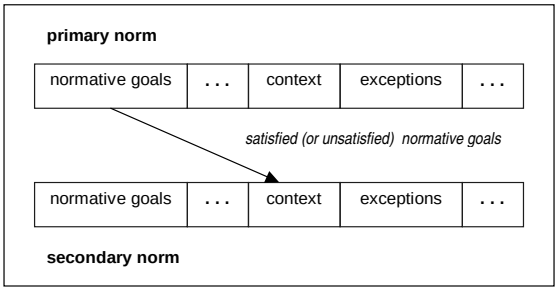



*Figure 1.* Interlocking Norm Structure

Formally, a norm is interlocked with another norm *by non-compliance* if, in the context of the secondary norm, an instance of the primary norm can be considered as violated. This means that when any addressee of a norm does not fulfill the norm, the corresponding interlocking norm will be triggered. The formal specification of this is given below, where $n_1$ represents the primary norm and $n_2$ is the secondary norm.

$$
\begin{array}{|l}
lockedbynoncompliance\_ : \mathbb{P}(Norm \times Norm) \\
\hline
\forall\, n_1, n_2 : Norm \bullet \\
\quad lockedbynoncompliance\ (n_1, n_2) \Leftrightarrow \\
\quad\quad (\exists\, ni : NormInstance \mid \\
\quad\quad isnorminstance\ (ni, n_1) \bullet \\
\quad\quad\quad \neg\ fulfilled\ (ni, n_2.context))
\end{array}
$$

Similarly, a norm is interlocked with another norm *by compliance* if, in the context of the secondary norm, an instance of the primary norm can be considered as fulfilled. Thus, any addressee of the norm that fulfills it will trigger the interlocking norm. The specification of this is given as follows.

$$
\begin{array}{|l}
lockedbycompliance\_ : \mathbb{P}(Norm \times Norm) \\
\hline
\forall\, n_1, n_2 : Norm \bullet \\
\quad lockedbycompliance\ (n_1, n_2) \Leftrightarrow \\
\quad\quad (\exists\, ni : NormInstance \mid \\
\quad\quad isnorminstance\ (ni, n_1) \bullet \\
\quad\quad\quad fulfilled\ (ni, n_2.context))
\end{array}
$$

Having the means to relate norms in this way allows us to model how the normative behaviour of agents that are addressees of a secondary norm is *influenced* by the normative behaviour of addressees of a primary norm. As can be observed, these relationships can be exploited to represent the so called *contrary-to-duty* norms.

## 4. Normative Multi-Agent Systems

Since norms are social concepts, they cannot be studied independently of the systems for which they are created and, consequently, an analysis of the normative aspects of social systems must be provided. Although social systems that are regulated by norms are different from one another, some general characteristics can be identified. They consist of a set of agents that are controlled by the same set of norms ranging from obligations and social commitments to social codes. However, whereas there are static systems in which all norms are defined in advance and agents in the system always comply with them (Boman, 1999; Shoham and Tennenholtz, 1995), a more realistic view of

these kinds of systems suggests that when *autonomous* agents are considered, neither can all norms be known in advance (since new conflicts among agents may emerge and, therefore, new norms may be needed), nor can compliance with norms be guaranteed (since agents can decide not to comply). We can say then, that systems regulated by norms must include mechanisms to deal with both the modification of norms and the unpredictable normative behaviour of autonomous agents. So, *normative multi-agent systems* have the following characteristics.

- *Membership*. Agents in a society must be able to deal with norms but, above all, they must recognise themselves as part of the system. This kind of social identification means that agents adopt the society norms and, by doing so, they show their willingness to comply with these norms.
- *Social Pressure*. Effective authority cannot be exerted if penalties or incentives are not applied when norms are either violated or complied with. However, this control must not be an agent's arbitrary decision, and although it is only exerted by some agents, it must be socially accepted.
- *Dynamism*. Normative systems are *dynamic* by nature. New norms are created and obsolete norms are abolished. Compliance or non-compliance with norms may activate other norms and, therefore, force other agents to act. Agents can either join or leave the system. The normative behaviour of agent members might be unexpected, and it may influence the behaviour of other agents.

Given these characteristics, we argue that multi-agent systems must include mechanisms to defend norms, to allow their modification, and to identify authorities. Moreover, their members must be agents able to deal with norms. Each one of these concepts is discussed in detail and formalised in (López y López and Luck, 2004), here, we present just a summary of them.

### 4.1. Normative Agents

The effectiveness of every structure of control relies on the capabilities of its members to recognise and follow its norms. However, given that agents are autonomous, the fulfillment of norms can never be taken for granted (López y López et al., 2002). We say that a *normative agent* is an agent whose behaviour is partly shaped by norms. They are able to deal with norms because they can represent, adopt, and comply with them. However, for autonomous agents, decisions to adopt or comply with norms are made on the basis of their own goals and motivations. That is, autonomous agents are not only able to *act on* norms but also they are able to *reason about* them. In what follows, all normative agents are considered as autonomous agents that have adopted some norms (*norms*) and, has decided which norms to comply with (*intended norms*) and which norms to reject (*rejected norms*). Although their normative behaviour is described in the next section, their representation is given now in the schema below.

$$
\begin{array}{|l}
\hline
\mathit{NormativeAgent} \\
\mathit{AutonomousAgent} \\
\mathit{norms}, \mathit{intented}, \mathit{rejected} : \mathbb{P}\, \mathit{Norm} \\
\hline
\mathit{intented} \subseteq \mathit{norms} \\
\mathit{rejected} \subseteq \mathit{norms} \\
\hline
\end{array}
$$

### 4.2. Enforcement and Reward Norms

Particularly interesting for this work are the norms triggered in order to punish offenders of other norms. We call them *enforcement norms* and their addressees are the *defenders* of a norm. These norms represent exerted social pressure because they specify not only who must apply the punishments, but also under which circumstances these punishments must be applied (Ross, 1968). That is, once the violation of a norm becomes identified by defenders, their duty is to start a process in which offender agents can be punished. For example, if there is an obligation to pay accommodation fees for all students in a university, there must also be a norm stating what hall managers must do when a student refuses to pay.

As can be seen, norms that enforce other norms are a special case of interlocking norms because besides being interlocked by non-compliance, the normative goals of the secondary norm must include every punishment of the primary norm. Figure 2 shows how the structures of both norms are related. By modelling enforcement norms in this way, we cause an offender's punishments to be consistent with a defender's responsibilities. Addressees of an *enforced* norm (i.e. the primary norm) know what could happen if the norm is not complied with, and addressees of an *enforcement* norm (i.e. the secondary norm) know what must be done in order to punish the offenders of another norm. Enforcement norms allow the authority of defenders to be clearly constrained.

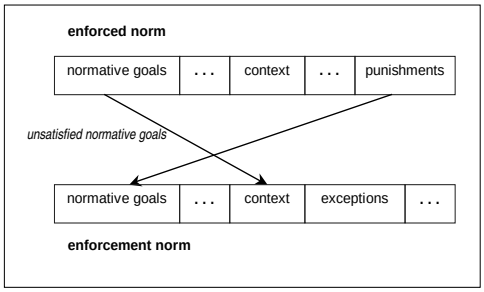



*Figure 2.* Enforcement Norm Structure

Formally, the relationship between a norm directed to control the behaviour of some agents and a norm directed at punishing the offenders of

such a norm can be defined as follows. A norm *enforces* another norm if the first norm is activated when the second is violated, and all punishments associated with the violated norm are part of the normative goals of the first. Every norm satisfying this property is known as an *enforcement* norm.

$$
\begin{array}{|l}
enforces\_ : \mathbb{P}(Norm \times Norm) \\
\hline
\forall\, n_1, n_2 : Norm \bullet enforces\ (n_1, n_2) \Leftrightarrow \\
\quad lockedbynoncompliance\ (n_2, n_1) \land \\
\quad n_2.punishments \subseteq n_1.normativegoals
\end{array}
$$

So far we have described some interlocking norms in terms of punishments because these are one of the more commonly used mechanisms to enforce compliance with norms. However, a similar analysis can be applied to interlocking norms corresponding to the process of rewarding members doing their duties. These norms must be interlocked by compliance and all the rewards included in the primary norm (rewarded norm) must be included in the normative goals of the secondary norm (reward norm). The relation between these norms is shown in Figure 3.

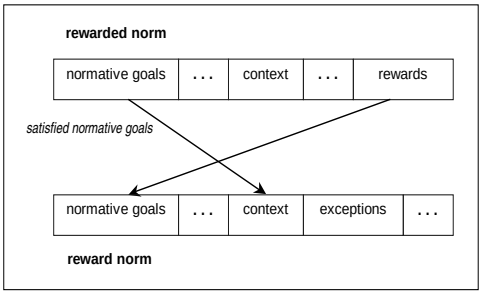



*Figure 3.* Reward Norm Structure

Formally, we say that a norm *encourages* compliance with another norm if the first norm is activated when the second norm becomes fulfilled, and the rewards associated with the fulfilled norm are part of the normative goals of the first norm. Every norm satisfying this property is known as a *reward* norm.

$$
\begin{array}{|l}
rewardnorm\_ : \mathbb{P}(Norm \times Norm) \\
\hline
\forall\, n_1, n_2 : Norm \bullet rewardnorm\ (n_1, n_2) \Leftrightarrow \\
\quad lockedbycompliance\ (n_2, n_1) \land \\
\quad n_2.rewards \subseteq n_1.normativegoals
\end{array}
$$

It is important to mention that this way of representing enforcement and reward norms can create an infinite chain of norms because we would

also have to define norms to apply when authorities or defenders do not comply with their obligations, either to punish those agents breaking rules or to reward those agents that fulfill their responsibilities (Ross, 1968). The decision of when to stop this interlocking of norms is left to the creator of norms. If a system requires it, the model (and formalisation) for enforcing and encouraging norms can be used recursively as necessary. There is nothing in the definition of the model itself to prevent this.

Both enforcement and reward norms acquire particular relevance in systems regulated by norms because the abilities to punish and reward must be restricted for use only by competent authorities (addressees of enforcement and reward norms). Otherwise, offenders might be punished twice or more if many agents take this as their responsibility. It could also be the case that selfish agents demand unjust punishments or that selfish offenders reject being punished. That is, conflicts of interest might emerge in a society if such responsibilities are given either to no one or to anyone. Only through enforcement and reward norms can agents become entitled to punish or reward other agents.

### 4.3. LEGISLATION NORMS

Norms are introduced into a society as a means to achieve social order. Some are intended to avoid conflicts between agents, others to allow the establishment of commitments, and others still to unify the behaviour of agents as a means of social identification. However, neither all conflicts nor all commitments can be anticipated. Consequently, there must exist the possibility of creating new norms (to solve unexpected and recurrent conflicts among agents), modifying existing ones (to increase their effectiveness), or even abolishing those that become obsolete. As above, these capabilities must be restricted to avoid conflicts of interest. That is, norms stating when actions to legislate are permitted must exist in a normative multi-agent system (Jones and Sergot, 1996). Formally, we say that a norm is a *legislation* norm if actions to issue and to abolish norms are permitted by this norm in the current environment. These constraints are specified below.

$$
\begin{array}{|l}
legislate\_ : \mathbb{P}(Norm \times EnvState) \\
\hline
\forall\, n : Norm;\ env : EnvState \bullet \\
legislate\ (n, env) \Leftrightarrow \\
\quad (\exists\, issuingnorms, abolishnorms : Action \bullet \\
\qquad permitted\ (issuingnorms, n, env) \lor \\
\qquad permitted\ (abolishnorms, n, env))
\end{array}
$$

### 4.4. NORMATIVE MULTI-AGENT SYSTEMS MODEL

A normative multi-agent system is formally represented in the $NormativeMAS$ schema. It comprises a set of normative agent members (i.e. agents able to

reason about norms) and a set of general norms that govern the behaviour of these agents ($generalnorms$). Norms issued to allow the creation and abolition of norms ($legislationnorms$) are also included. There are also norms dedicated to enforcing other norms ($enforcenorms$) and norms directed to encouraging compliance with norms through rewards ($rewardnorms$). Legislation, enforcement and reward norms are better discussed in (López y López and Luck, 2004). The current state of the environment is represented by the variable $environment$. Constraints over these components are imposed as follows. Although it is possible that agents do not know all the norms in the system, it is always expected that they at least adopt some norms, represented by the first predicate. The second predicate makes explicit that addressees of norms must be members of the system. Thus, addressee agents of every norm must be included in the set of member agents because it does not make any sense to have norms addressed to nonexistent agents. The last three predicates respectively describe the structure of enforcement, reward and legislation norms. Notice that whereas every enforcement norm must have a norm to enforce, not every norm may have a corresponding enforcement norm, in which case no one in the society is legally entitled to punish an agent that does not fulfill such a norm.

$$
\begin{array}{|l}
\hline
NormativeMAS \\
members : \mathbb{P}\, NormativeAgent \\
generalnorms, legislationnorms : \mathbb{P}\, Norm \\
enforcenorms, rewardnorms : \mathbb{P}\, Norm \\
environment : EnvState \\
\hline
\forall\, ag : members \bullet \\
\quad ag.norms \cap generalnorms \neq \varnothing \\
\forall\, sn : generalnorms \bullet \\
\quad sn.addressees \subseteq members \\
\forall\, en : enforcenorms \bullet \\
\quad (\exists\, n : generalnorms \bullet enforces\ (en, n)) \\
\forall\, rn : rewardnorms \bullet \\
\quad (\exists\, n : generalnorms \bullet rewardnorm\ (rn, n)) \\
\forall\, ln : legislationnorms \bullet \\
\quad legislate\ (ln, environment) \\
\hline
\end{array}
$$

## 4.5. NORMATIVE ROLES

Defining normative multi-agent systems in this way allows the identification of the *authorities* of the system as formalised in the $AuthoritiesNMAS$ schema. The set of agents that are entitled to create, modify, or abolish norms is called *legislators*. No other members of the society are endowed with this authority, and generally they are either elected or imposed by other agents. *Defender* agents are directly responsible for the application of punishments

when norms are violated. That is, their main responsibility is to monitor compliance with norms in order to detect transgressions. Moreover, they can also warn agents by advertising the bad consequences of being rebellious. By contrast, *promoter* agents are those whose responsibilities include rewarding compliant addressees. These agents also monitor compliance with norms in order to determine when rewards must be given, and instead of *enforcing* compliance with norms, they simply *encourage* it.

$$
\begin{array}{|l}
\hline
AuthoritiesNMAS \\
\hline
NormativeMAS \\
legislators : \mathbb{P}\, NormativeAgent \\
defenders : \mathbb{P}\, NormativeAgent \\
promoters : \mathbb{P}\, NormativeAgent \\
\hline
\forall\, lg : legislators \bullet (\exists\, l : legislationnorms \bullet \\
\quad lg \in l.addressees) \\
\forall\, df : defenders \bullet (\exists\, e : enforcenorms \bullet \\
\quad df \in e.addressees) \\
\forall\, pm : promoters \bullet (\exists\, r : rewardnorms \bullet \\
\quad pm \in r.addressees) \\
\hline
\end{array}
$$

## 5. Autonomous Normative Reasoning

Whereas agents that always comply with norms are important for the design of societies in which total control is needed (Boman, 1999; Shoham and Tennenholtz, 1995), agents that can decide on the basis of their own goals and motivations whether to comply with them are important for the design of dynamic systems in which agents act on behalf of different users and, while satisfying their own goals, are able to join a society and cooperate with other agents. Autonomous norm reasoning is important to address those situations in which an agent's goals conflict with the norms that control its behaviour inside a society. Agents that deliberate about norms are also needed in systems in which unforseen events might occur, and in those situations in which agents are faced with conflicting norms, and they have to choose between them. It should be clear that violation of norms is, sometimes, justified. To describe *normative reasoning*, therefore, we have to explain not only what might motivate an agent to adopt, dismiss or complying with a norm, but also the way in which this decision affects its goals. In consequence we propose three different normative reasoning processes: one for agents to decide whether to adopt a norm (*the norm adoption process*), another to decide whether to comply with a norm (*the norm deliberation process*), and the other to update the goals, and therefore the intentions of agents accordingly (*the norm compliance process*). All these processes must take into account not

only the goals and motivations of agents, but also the mechanisms of the society to avoid violation of norms such as rewards and punishments. Thus, agents consider the so called *social pressure of norms* before making any decision.

### 5.1. THE NORM ADOPTION PROCESS

The *norm adoption* process can be better defined as the process through which agents recognise their responsibilities towards other agents by internalising the norms that specify these responsibilities. Thus, agents adopt the norms of a society either once they have decided to join it or in the case a new norm is issued while they are still there. For autonomous agents to join and stay in a society the *social satisfaction condition* must hold (López y López et al., 2004). An agent considers this condition as satisfied if, although some of its goals become hindered by its *responsibilities*, its important goals can still be satisfied. Thus, we consider that the following conditions must be satisfied for agents to adopt a norm: the agent must recognise itself as an addressee of the norm; the norm must not already be adopted; the norm must have been issued by a recognised authority; and the agent must have reasons to stay in the society. Notice that to adopt a norm as an end, only the first three conditions are needed, whereas the last condition is an indicator that the decision to adopt a norm is made in an autonomous way. Due to space constraints, the $NormAdoption$ schema only formalises the first three conditions but details of the fourth condition can be found elsewhere (López y López et al., 2004).

$$
\begin{array}{|l}
\hline
NormAdoption \\
\Delta NormativeAgent \\
new? : Norm \\
issuer?, self : NormativeAgent \\
authorities : \mathbb{P}\, NormativeAgent \\
issuedby : \mathbb{P}(Norm \times NormativeAgent) \\
\hline
self \in new?.addressees \\
new? \notin norms \\
(new?,\ issuer?) \in issuedby \Leftrightarrow \\
\quad issuer? \in authorities \\
norms' = norms \cup \{new?\} \\
\hline
\end{array}
$$

### 5.2. THE NORM DELIBERATION PROCESS

To comply with the norm, agents assess two things: the goals that might be hindered by satisfying the normative goals, and the goals that might benefit from the associated rewards. By contrast, to reject a norm, agents evaluate the damaging effects of punishments (i.e. the goals hindered due to the satisfaction of the goals associated with punishments.) Since the satisfaction of some of their goals might be prevented in both cases, agents use the *importance* of

their goals to make these decisions. The importance of goals is a term related to an agent's motivations (d'Inverno and Luck, 2003) in such a way that the more motivated a goal the higher its importance. Thus, to deliberate about a norm, agents follow these steps.

- A set of *active* norms is selected from the set of adopted norms (norm instances). Active norms are those that agents believe must be complied with in the current state, which is not an exception state (i.e. those norms for which the context matches the beliefs of the agent).
- The agent divides active norms into *non-conflicticting* and *conflicting* norms. An active norm is *non-conflicting* if its compliance does not cause any conflict with one of the agent's current goals. Thus, no goals of the addressee agent are hindered by satisfying the normative goals of the norm. By contrast, an active norm is *conflicting* if its fulfillment hinders any of the agent's goals.
- For each one of these sets of norms, the agent must decide which one to comply with. Details of different ways to select the norms to be intended or rejected are given in (López y López et al., 2002). Here only one of them is explained later on. After norm deliberation, the set of intended norms consists of those conflicting and non-conflicting norms that are accepted to be complied with by the agent, and the set of rejected norms consists of all conflicting and non-conflicting norms that are rejected by the agent.

$$
\begin{array}{|l}
\hline
NormAgentState \\
NormativeAgent \\
activenorms, conflicting\_ : \mathbb{P}\, Norm \\
\hline
\forall\, n : activenorms \bullet conflicting\ n \Leftrightarrow \\
\quad hinder(goals, n.ngoals) \neq \varnothing \\
activenorms \subseteq norms \\
\forall\, an : activenorms \bullet \\
\quad logcon\ (beliefs, an.context) \\
activenorms = intended \cup rejected \\
hinder(goals, normgoals\ intended) = \varnothing \\
benefit(goals, rewardgoals\ intended) \\
\quad \cap goals = \varnothing \\
hinder(goals, punishgoals\ rejected) = \varnothing \\
\hline
\end{array}
$$

The state of an agent that has selected the norms it is keen to fulfill is formally represented in the $NormAgentState$ schema above. This represents a normative agent with a variable representing the sets of $active$ norms at a particular point of time. The $conflicting$ predicate holds for a norm if and only

if its normative goals conflict ($hinder$) with any of the agent's current goals. The next three predicates state that active norms are the subset of adopted norms that the agent believes must be complied with in the current state and that, the set of active norms has already been assessed and divided into norms to intend and norms to reject. The state of an agent is consistent in that its current goals do not conflict with the intended norms and, consequently, no normative goal must be in conflict with current goals. Moreover, since rewards benefit the achievement of some goals, so that agents do not have to work on their satisfaction because someone else does, these goals must not be part of the goals of an agent. The final predicate states that punishments must be accepted and, therefore, none of the goals of an agent must hinder them.

For a norm to be intended, some constraints must be fulfilled. First, the agent must be an addressee of the norm. Then, the norm must be an adopted and currently active norm, and it must not be already intended. In addition, the agent must believe that it is not in an *exception* state and, therefore, it must comply with the norm. Formally, the process to accept a single norm as input ($new?$) to be complied with is specified in the $NormIntend$ schema. The first five predicates represent the constraints on the agent and the norm as described above. The sixth predicate represents the addition of the accepted norm to the set of intended norms and the final predicate represents the set of rejected norms remains the same.

$$
\begin{array}{|l}
\hline
NormIntend \\
new? : Norm \\
\Delta NormAgentState \\
\hline
self \in new?.addressees \\
new? \in norms \\
new? \in activenorms \\
new? \notin intended \\
\neg\ logcon(beliefs, new?.exceptions) \\
intended' = intended \cup \{new?\} \\
rejected' = rejected \\
\hline
\end{array}
$$

The process to reject a norm ($NormReject$) can be defined similarly. Now, there are different ways to select the norms to be intended or rejected as explained in (López y López et al., 2002). Here, we describe what is called a *pressured* strategy where an agent fulfills a norm only in the case that one of its goals is threatened by punishments. That is, agents are *pressured* to obey norms through the application of punishments that might hinder some of their important goals. In this situation, the agent faces four different cases.

1. The norm is a non-conflicting norm and some goals are hindered by its punishments.

2. The norm is a non-conflicting norm and there are no goals hindered by its punishments.

3. The norm is a conflicting norm and the goals hindered by its normative goals are less important than the goals hindered by its punishments.

4. The norm is a conflicting norm and the goals hindered by its normative goals are more important than the goals hindered by its punishments.

The first case represents the situation in which, by complying with a norm, an agent does not put at risk any of its goals (because the norm is non-conflicting), but if the agent decides not to fulfill it, some of its goals could be unsatisfied due to punishments. Consequently, fulfilling a norm is the best decision for this kind of agent. To formalise this, we use the $NormIntend$ operation schema to accept complying with the norm, and we add two predicates to specify that this strategy is applied to non-conflicting norms whose punishments hinder some goals.

$PressuredNCComply$

$NormIntend$

---

$\neg\ conflicting\ new?$

$hinder(goals, new?.punishments) \neq \varnothing$

In the second case, by contrast, since punishments do not affect an agent's goals, it does not make any sense to comply with the norm, so it must be rejected. Formally, the $NormReject$ operation schema is used when the norm is non-conflicting (first predicate) and its associated punishments do not hinder any existing goals (second predicate).

$PressuredNCReject$

$NormReject$

---

$\neg\ conflicting\ new?$

$hinder(goals, new?.punishments) = \varnothing$

According to our definition, a conflicting norm is a norm whose normative goals hinder an agent's goals. In this situation, agents comply with the norm at the expense of existing goals only if what they can lose through punishments is more important than what they can lose by complying with the norm. Formally, a conflicting norm is intended if the goals that could be hindered by punishments ($hps$) are more important than the set of existing goals hindered by normative goals ($hngs$). This is represented in the $PressuredCComply$ schema where the $importance$ function uses the motivations associated with the set of goals to find the importance of goals.

*PressuredCComply*
*NormIntend*

$conflicting\ new?$
**let** $hps == hinder(goals,$
$new?.punishments) \bullet$
**let** $hngs == hinder(goals, new?.ngoals) \bullet$
$importance\ (motivations, hps) >$
$importance\ (motivations, hngs)$

However, if the goals hindered by normative goals are more important than the goals hindered by punishment, agents prefer to face such punishments for the sake of their important goals and, therefore, the norm is rejected. Formally, a conflicting norm is rejected by using the $NormReject$ operation schema if the goals hindered by its punishments ($hps$) are less important than the goals hindered by its normative goals ($hngs$).

*PressuredCReject*
*NormReject*

$conflicting\ new?$
**let** $hps == hinder(goals, new?.punishments) \bullet$
**let** $hngs == hinder(goals, new?.ngoals) \bullet$
$importance\ (motivations, hps) \leq$
$importance\ (motivations, hngs)$

All these cases are illustrated in Figure 4

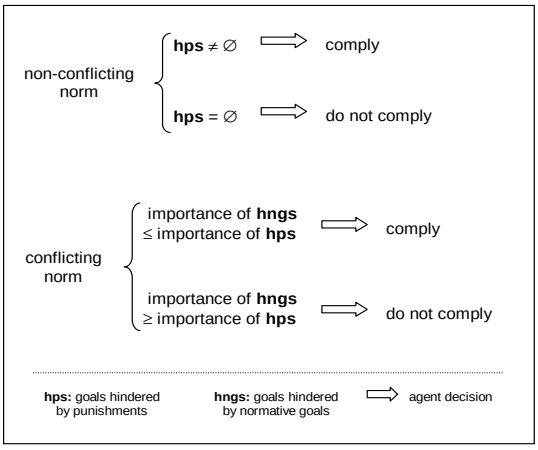



*Figure 4.* Pressured Norm Compliance

### 5.3. THE NORM COMPLIANCE PROCESS

Once agents take a decision about which norms to fulfill, a process of *norm compliance* must be started in order to update an agent's goals in accordance with the decisions it has made. An agent's goals are affected in different ways, depending on whether the norm is intended or rejected. The cases can be listed as follows.

- All normative goals (*ngs*) of an intended norm must be added to the set of goals because the agent has decided to comply with it.
- Some goals (*hngs)* are hindered by the normative goals of an intended norm. These goals can no longer be achieved because the agent prefers to comply with the norm and, consequently, this set of goals must be removed from the agent's goals.
- Some goals (*brs*) benefit from the rewards of an intended norm. Rewards contribute to the satisfaction of these goals without the agent having to make any extra effort. As a result, those goals that benefit from rewards must no longer be considered by the agent to be satisfied, and must be removed from the set of goals.
- Rejected norms only affect the set of goals hindered by the associated punishments (*hps*). This set of goals must be removed; this is the way in which normative agents accept the consequences of their decisions.

The process to comply with the norms an agent has decided to fulfill is specified in the $NormComply$ schema. Through this process, the set of goals is updated according to our discussion above.

$$
\begin{array}{|l}
\hline
NormComply \\
\Delta NormAgentState \\
\hline
\textbf{let}\ ngs == \bigcup\{gs : \mathbb{P}\ Goal \mid \\
\quad (\exists\, n : intended \bullet gs = n.ngoals)\} \bullet \\
\textbf{let}\ hngs == \bigcup\{gs : \mathbb{P}\ Goal \mid (\exists\, n : intended \bullet \\
\quad gs = hinder\ (goals, n.ngoals))\} \bullet \\
\textbf{let}\ brs == \bigcup\{gs : \mathbb{P}\ Goal \mid (\exists\, n : intended \bullet \\
\quad gs = benefit(goals, n.rewards))\} \bullet \\
\textbf{let}\ hps == \bigcup\{gs : \mathbb{P}\ Goal \mid (\exists\, n : rejected \bullet \\
\quad gs = hinder\ (goals, n.punishments))\} \bullet \\
(\ goals' = (goals \cup ngs) \setminus \\
\qquad\qquad (hngs \cup brs \cup hps)) \\
\hline
\end{array}
$$

However, to make the model simple, we assume that punishments are always applied, and rewards are always given, though the possibility exists that agents never become either punished or rewarded. In addition, note that the set of goals hindered by normative goals can be empty if the norm being

considered is a non-conflicting norm, and goals hindered by punishments or goals that benefit from rewards can be empty if a norm does not include any of them. After norm compliance, the goals are updated and, consequently, the intentions of agents might change.

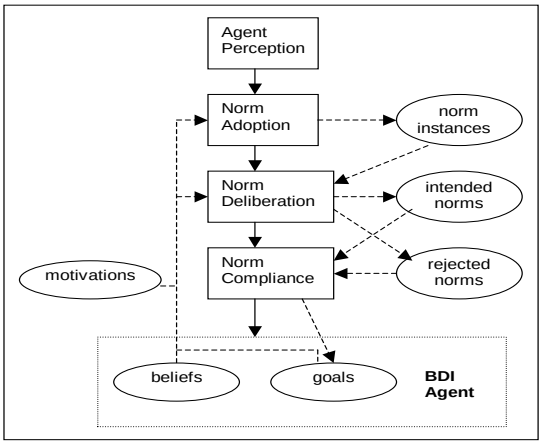



*Figure 5.* Normative Reasoning Processes and Outcomes

Figure 5 summarises the three processes already mentioned and the new mental attitudes that appear as a consequence of normative reasoning. Continuous arrows represent the control flow and dashed lines represent data flow.

## 6. Conclusions

In this paper, we have presented a normative framework which, besides providing the means to computationally represent many normative concepts, can be used to give a better understanding of norms and normative agent behaviour. The framework explains not only the role that norms play in a society but also the elements that constitute a norm and that, in turn, can be used by agents when decisions concerning norms must be taken. In contrast to other proposals, our normative framework has been built upon the idea of *autonomy* of agents. That is, it is intended to be used by agents that reason about why norms must be adopted, and why an adopted norm must be complied with. Our framework consists of three main components: a canonical model of norms, a model of normative multi-agent systems and a model of normative autonomous agents.

The model of norms differs from others (Boman, 1999; Shoham and Tennenholtz, 1995; Tuomela, 1995) in the way in which patterns of behaviour are prescribed. To describe the pattern of behaviour prescribed by a norm,

other models use actions, so that agents are told what exactly they must do. By contrast, we use normative goals, which is an idea more compatible with autonomous agents whose behaviour is driven by goals. Agents can choose the way to satisfy the normative goals, instead of being told exactly how it must be done. Our work also emphasises that all norms can be represented by using similar components, and that they are analysed by agents in similar ways. However, what makes one norm different from another is the way in which norms are created, how long they are valid, and the reasons agents have to adopt them. These factors enable norms to be divided into categories such as obligations and prohibitions, social commitments and social codes.

A collateral result of our work is the proposed model for interlocking norms. These relations between norms have already been mentioned in several papers, especially from philosophical and legal perspectives (Ross, 1968), but no ways to model them have been provided. Dignum's concept of authorisations (Dignum, 1999) attempts to describe norms activated when others are not fulfilled; however, his idea and models are incomplete. We claim that this form of representing connections between norms can be used not only to represent enforcement and reward norms, but also to represent things as complex as contracts and deals among agents. Enforcement and reward norms are a special case of interlocking norms, they prescribe what must be done in the cases in which a norm becomes either fulfilled or unfulfilled. Although this way of relating norms allows the representation of contrary-to-duty norms it does not eliminate those pitfalls related to the contrary-to-duty or the dilemma paradoxes (van der Torre and Tan, 2000). How to deal with these issues is beyond the scope of this paper.

In contrast to current models of systems regulated by norms (Balzer and Tuomela, 2001; Dignum, 2004; Dignum and Dignum, 2001; Esteva et al., 2001; Shoham and Tennenholtz, 1995) in which no distinction among norms is made, our work emphasises that besides the general norms of the system, at least three kinds of norms are needed, namely norms to legislate, to punish, and to reward other agents. By making this differentiation, agents are able to determine when an issued norm is valid, when an entitled agent can apply a punishment, and who is responsible for giving rewards. In addition, order is imposed on agents responsible for the normative behaviour of other agents, because their authority is defined by the norms that entitle them to exert social pressure. Roles for *legislators*, *defenders*, and *promoters* of norms become easily identified as a consequence of the different kinds of norms considered. Thus, in this framework, the authority of agents is always supported and constrained by norms. Our framework does not intent to override other frameworks such as those already mentioned above rather it intends to complement them. Current work is doing to use the framework to implement normative agents, virtual communities of service provider agents, and to represent contracts and contract reasoning.

## References


Axelrod, R. (1986). An evolutionary approach to norms. *The American Political Science Review*, 80(4):1095–1111.

Balzer, W. and Tuomela, R. (2001). Social institutions, norms and practices. In Dellarocas, C. and Conte, R., editors, *Social Order in Multi-Agent Systems*, pages 161–180. Kluwer Academic Publishers.

Barbuceanu, M., Gray, T., and Mankovski, S. (1999). The role of obligations in multiagent coordination. *Applied Artificial Intelligence*, 13(1/2):11–38.

Bicchieri, C. (1990). Norms of cooperation. *Ethics*, 100(4):838–861.

Boella, G. and Lesmo, L. (2001). Deliberative normative agents. In Dellarocas, C. and Conte, R., editors, *Social Order in Multi-Agent Systems*, pages 85–110. Kluwer Academic Publishers.

Boman, M. (1999). Norms in artificial decision making. *Artificial Intelligence and Law*, 7(1):17–35.

Castelfranchi, C., Conte, R., and Paolucci, M. (1998). Normative reputation and the cost of compliance. *Journal of Artificial Societies and Social Simulation*, 1(3).

Castelfranchi, C., Dignum, F., Jonker, C., and Treur, J. (2000). Deliberative normative agents: Principles and architecture. In Jennings, N. and Lesperance, Y., editors, *Intelligent Agents VI (ATAL'99)*, LNAI 1757, pages 206–220. Springer-Verlag.

Conte, R. and Castelfranchi, C. (1995). *Cognitive and Social Action*. UCL Press.

Conte, R., Castelfranchi, C., and Dignum, F. (1999a). Autonomous norm-acceptance. In Müller, J., Singh, M., and Rao, A., editors, *Intelligent Agents V (ATAL'98)*, LNAI 1555, pages 319–333. Springer-Verlag.

Conte, R. and Dellarocas, C. (2001). Social order in info societies: An old challenge for innovation. In Dellarocas, C. and Conte, R., editors, *Social Order in Multi-Agent Systems*, pages 1–15. Kluwer Academic Publishers.

Conte, R., Falcone, R., and Sartor, G. (1999b). Agents and norms: How to fill the gap? *Artificial Intelligence and Law*, 7(1):1–15.

Dellarocas, C. and Klein, M. (2001). Contractual agent societies: Negotiated shared context and social control in open multi-agent systems. In Dellarocas, C. and Conte, R., editors, *Social Order in Multi-Agent Systems*, pages 113–133. Kluwer Academic Publishers.

Dignum, F. (1999). Autonomous agents with norms. *Artificial Intelligence and Law*, 7(1):69–79.

Dignum, F., Morley, D., Sonenberg, E., and Cavendon, L. (2000). Towards socially sophisticated BDI agents. In Durfee, E. H., editor, *Proceedings on the Fourth International Conference on Multi-Agent Systems (ICMAS-00)*, pages 111–118. IEEE Computer Society.

Dignum, V. (2004). *A Model for Organizational Interaction*. Phd, University of Utrecht, The Netherlands.

Dignum, V. and Dignum, F. (2001). Modelling agent societies: Coordination frameworks and institutions. In Brazdil, P. and Jorge, A., editors, *Progress in Artificial Intelligence Knowledge Extraction, Multi-agent Systems, Logic Programming, and Constraint Solving*, LNAI 2258, pages 191–204. Springer-Verlag.

d'Inverno, M. and Luck, M. (2003). *Understanding Agent Systems*. Springer-Verlag, second edition.

Esteva, M., Padget, J., and Sierra, C. (2001). Formalizing a language for institutions and norms. In Meyer, J. and Tambe, M., editors, *Intelligent Agents VIII (ATAL'01)*, LNAI 2333, pages 348–366. Springer-Verlag.

Hashimoto, T. and Egashira, S. (2001). Formation of social norms in communicating agents with cognitive frameworks. *Systems Science and Complexity*, 14(1):54–74.

Jones, A. and Sergot, M. (1996). A formal characterisation of institutionalised power. *Logic Journal of the IGPL*, 4(3):429–445.

López y López, F. (2003). *Social Powers and Norms: Impact on Agent Behaviour*. Phd, University of Southampton, England.

López y López, F. and Luck, M. (2003). Modelling norms for autonomous agents. In Chávez, E., Favela, J., Mejía, M., and Oliart, A., editors, *Proceedings of the Fourth Mexican International Conference on Computer Science (ENC'03)*, pages 238–245. IEEE Computer Society.

López y López, F. and Luck, M. (2004). A model of normative multi-agent systems and dynamic relationships. In Lindemann, G., Moldt, D., and Paolucci, M., editors, *Regulated Agent-Based Social Systems*, LNAI 2934, pages 259–280. Springer-Verlag.

López y López, F., Luck, M., and d'Inverno, M. (2002). Constraining autonomy through norms. In Castelfranchi, C. and Johnson, W., editors, *Proceedings of The First International Joint Conference on Autonomous Agents and Multi Agent Systems AAMAS'02*, pages 674–681. ACM Press.

López y López, F., Luck, M., and d'Inverno, M. (2004). Normative agent reasoning in dynamic societies. In Jennings, N., Sierra, C., Sonenberg, L., and Tambe, L., editors, *Proceedings of The Third International Joint Conference on Autonomous Agents and Multi Agent Systems AAMAS'04*, pages 730–737. ACM Press.

Luck, M., McBurney, P., and Preist, C. (2003). *Agent Technology: Enabling Next Generation Computing (A Roadmap for Agent Based Computing)*. AgentLink.

Norman, T., Sierra, C., and Jennings, N. (1998). Rights and commitments in multi-agent agreements. In Demazeau, Y., editor, *Proceedings of the Third International Conference on Multi-Agent Systems (ICMAS-98)*, pages 222–229. IEEE Computer Society Press.

Ross, A. (1968). *Directives and Norms*. Routledge and Kegan Paul Ltd.

Sergot, M. (1999). Normative positions. In McNamara, P. and Prakken, H., editors, *Norms, Logics and Information Systems*, pages 289–308. IOS Press.

Shoham, Y. and Tennenholtz, M. (1995). On social laws for artificial agent societies: Off-line design. *Artificial Intelligence*, 73(1-2):231–252.

Singh, M. (1999). An ontology for commitments in multiagent systems: Toward a unification of normative concepts. *Artificial Intelligence and Law*, 7(1):97–113.

Spivey, J. M. (1992). *The Z Notation: A Reference Manual*. Prentice-Hall.

Tuomela, R. (1995). *The Importance of Us: A Philosophical Study of Basic Social Norms*. Stanford University Press.

Tuomela, R. and Bonnevier-Toumela, M. (1995). Norms and agreements. *European Journal of Law, Philosophy and Computer Sience*, 5:41–46.

Ullmann-Margalit, E. (1977). *The Emergence of Norms*. Oxford University Press.

van der Torre, L. and Tan, Y. (1999a). Contrary-to-duty reasoning with preference-based dyadic obligations. *Annals of Mathematics and Artificial Intelligence*, 27(1-4):49–78.

van der Torre, L. and Tan, Y. (1999b). Rights, duties and commitments between agents. In *Proceedings of the Sixteenth International Joint Conference on Artificial Intelligence*, pages 1239–1246.

van der Torre, L. and Tan, Y. (2000). Dynamic normative reasoning under uncertainty. In Smets, P. and Gabbay, D., editors, *Agents, Reasoning and Dynamics*. Kluwer.

Walker, A. and Wooldridge, M. (1995). Understanding the emergence of conventions in multi-agent systems. In Lesser, V. and Gasser, L., editors, *Proceedings of the First International Conference on Multi-Agent Systems (ICMAS'95)*, pages 384–389. AAAI Press/MIT Press.

Wieringa, R., Dignum, F., Meyer, J., and Kuiper, R. (1996). A modal approach to intentions, commitments and obligations: Intention plus commitment yields obligation. In Brown, M. and Carmo, J., editors, *Deontic Logic, Agency and Normative Systems*, pages 80–97. Springer-Verlag.

# A Normative Multi-Agent Systems Approach to the Use of Conviviality for Digital Cities

Patrice Caire

University of Luxembourg, Computer Science Department
L-1359, Luxembourg, 6, Rue Richard Coudenhove-Kalergi, Luxembourg
patrice.caire@uni.lu

**Abstract.** Conviviality is a mechanism to reinforce social cohesion and a tool to reduce mis-coordination between individuals, groups and institutions in web communities, for example in digital cities. We use a two-fold definition of conviviality as a condition for social interactions and an instrument for the internal regulation of social systems. In this paper we discuss the use of normative multi-agent systems to analyze the use of conviviality for digital cities, by contrasting norms for conviviality with legal and institutional norms in digital cities. We show the role of the distinction among various kinds of norms, the explicit representation of norms, the violability of norms, the dynamics of norms and the creation of norms in the context of conviviality.

**Keywords.** Conviviality, multi-agent systems, normative systems, social computing, digital cities.

## 1 Introduction

The role of norms for conviviality is a condition for social interactions and an instrument for the internal regulation of social systems [1]. For example, in digital cities "government regulations extend laws with specific guidance to corporate and public actions" [2].

In this paper we raise the following question: how can normative multi-agent systems be used to model conviviality? We approach this question focussing on conviviality in digital cities, and by contrasting the use of normative multi-agent systems for conviviality with legal and institutional norms in digital cities.

Our main question breaks down into the following research questions: What are digital cities, what are normative multi-agent systems, what is conviviality and finally, can norms be applied to conviviality?

The layout of this paper follows these sub-questions. In section 2 we give a brief overview on digital cities, in section 3 we explain norms in regards to the legal and institutional aspects of digital cities, in section 4 we present a literature survey on the notion of conviviality and in section 5 we examine the use of norms for conviviality.

Dagstuhl Seminar Proceedings 07122
Normative Multi-agent Systems
http://drops.dagstuhl.de/opus/volltexte/2007/906

## 2 Brief Overview on Digital Cities

There are many ways to define digital cities. "They can be seen as a local social information infrastructure, providing information over the real city to locals and of course to visitors of the real city. The digital city can also be approached as a communication medium, influencing the personal networks of inhabitants of a digital neighborhood. Another view is the digital city as a tool to improve local democracy and participation, in fact the basic idea behind the digital city in Amsterdam. Fourth, we can characterize the digital city as a free space to experience and experiment with cyberspace. Finally, the digital city can be seen as a practical resource for the organization of every day life. One can think of local electronic commerce, and the provision of online public services as a support of local economic activities. However, the digital city may also become an experiment with new forms of solving problems and coordinating social life. Where currently most activities are coordinated by the market or by the state, the digital city may become a tool that enables people to do things by mobilizing the available local resources, using existing and emerging social networks"[3].

### 2.1 Digital Cities Categories and Goals

There are five Broad Categories of digital cities: Non-profit grass-roots community initiatives such as Amsterdam DDS-De Digitale Stad, municipal information and communication networks such as E-governments, commercial city oriented web sites, for instance AOL Digital Cities and Microsoft CitySearch, virtual environment for virtual communities ahd communities of interest such as the scientific community and finally social ICT experiments, often in less privileged neighborhoods.

The main goals of digital cities are to create public space, exchange social information, explore vertical markets and innovate with next generation networks. Today their principal objective is to transform and modernize the local administration to improve the level and quality of life of the population at both individual and community levels.

### 2.2 American Community Networks

Digital cities started in the US with american community networks, inspired by a tradition of community-centered, grass-roots engagements that emphasized freedom of speech and activism. Their original goal was to create a virtual information space at first non-geographically bound for example the WELL, "Whole Earth 'Lectronic Link", but that subsequently evolved into geographically bound for example Blacksburg' Electronic Village. Their main challenges were first, the lack of synergy between community networks, private companies and administrations and second, the competition between profit and non-profit organizations.

*Case study 1: Blacksburg Electronic Village* was built in 91 as a consortium lead by universities, among which Virginia Tech University, by regional companies such as Bell Atlantic and local authorities. It was a high profile project

but with very little community involvement to the vision. It was constructed from a technological point of view and the first project of the kind with web interface. It rapidly grew until 95 then its activity decreased due to fundamental disagreement between all the partners' expectations. The companies looked for revenues elsewhere and universities stopped providing internet to non-university members. It is still active today but with very local focus on community use of technology and learning.

*Case study 2: Seattle Community Network* emerged in 92 as part of the "Computer Professionals for Social Responsibilities" group's civil activities. It was first hosted on a donated Intel 386 running a donated copy of BSDI UNIX operating system, using FreePort (Cleveland FreeNet text based) user interface software. It was lead by citizens and grew in size by cooperating with regional libraries and offering free network access and services to all: email, homepages etc. Due to continuous financial problems and competition with commercial portals the activity decreased to its current reduced level and provides free public-access network. Interestingly, among the more recent grass-roots activities is the emergence of the Seattle Community Wireless network that creates a broadband wireless metropolitan area network.

### 2.3 European Digital Cities

The goal of European digital cities is to integrate and coordinate private, public and voluntary sectors towards better regional and local information system. The European Community first launched the Telecities program in 93 later evolved with large scope programs and projects such as Eurocities, Intelcities and e-Agora. For the European Community, the goal is to share ideas and technologies between all the cities to strengthen the European partnership. For cities, the goal is to use information and communication technologies to resolve social, economic and regional development issues and improve the quality of social services. Their characteristics today are to be networks generated within and for specific regions, to form complex communities based on collaborations between citizens, universities, city administrations and private companies, and to emphasize social inclusion. Their main challenge is the difficulty to integrate grass-roots communities and commercial point of views which appears in the relatively slow commercialization of services and information.

*Case study 1: Amsterdam Digital City, DDS,* started in 94 as a grass-roots initiative and evolved into a non-profit organization with government support and the participation of private companies. The goals of Digital Amsterdam were to support community activities and local economy, encourage political discourse by linking citizens to the administration and innovate. Its very successful interface of squares and cafes and interactive public debates inspired many other digital cities, among others digital Bristol. The issues that caused its downfall were persistent technical problems and the initial lack of common understanding and vision between the stakeholders. Digital Amsterdam exploited all the early Internet possibilities such as USENET, IRC, GOPHER, MUDs, MOOs, Telnets and Free-nets.

*Case study 2: Virtual Helsinki* started in 95 as a powerful consortium of Telecom (Nokia, Elisa), the city of Helsinki, private companies such as IBM and local universities but did not include any grass-roots community nor voluntary services. It had three goals: Technological advances with for instance the use of ISDN and Video on Demand (95), DSL, Ethernet, ADSL (97), IP based Video conferencing (98), ISDN video telephony, 3-D mapping of Helsinki (99). Digital Helsinki has been highly profitable and socially relevant with citizens' participation and contribution to social cohesion. Its projects of using avatars for citizens inspired the Habbo community.

*Case study 3: Bologna 'Iperbole' and Issy-les-Moulineaux.* Started in 94, Bologna's great innovation was to create an open space for citizen groups to publish information and engage in debates with public officials while Issy-les-Moulineaux in France, started in 96, developped a very successful one-stop administration with online live interaction of citizens in town meetings, interactive map, and so on.

### 2.4 Asian City Informatization

Asian digital cities, actually called city informatization, emerged as government initiatives. Their goal is to develop their country through technological innovation. Singapore initiated the process in 92 and launched in 96, Japan in 94, Korea 95 and Malaysia 96. There were attempts to integrate grass-roots activities and university driven projects in 99 with Digital Kyoto and Shanghai but the greatest challenge still remains their top-down approach based on administration activity.

### 2.5 Commercial Portals

Commercial portals started as local portals run by private companies, such as phone or web companies and airlines, competing with each other. Nowadays, global companies such as AOL and Microsoft offer city guides with services: shopping, entertainment, some local information and maps. Their general trend is to provide easy to find and search information, good maintenance of systems and frequent updates. They are effective in Asia, where they complement government agencies, but limited in scope by their top-down controlled and selected content, lack of two-way interaction with users and main purpose, e.g. advertising.

### 2.6 Digital Cities Models and Future Growth

As yet, no one model has been identified. In the US for-profit businesses and non-profit organizations co-exist and compete, in EU the attempts are to coordinate administrations, companies and citizens while Asia pursues government directed growth. Ultimately, digital cities need to deal with the same complexity as real cities to attract and retain usage, and to function as entities that augment their physical counterparts.

The main goals of digital cities consist to help close the digital divide be it geographic, with access everywhere, or social, with access for all, accelerate economic development, and make cities' governments more efficient and accessible. The means used by digital cities to achieve these goals are: Pluralism and participation, combined multi-disciplinary approaches, synergy between their three constitutive elements: administration, companies and citizens and finally a shared vision between all stakeholders.

The success factors of digital cities consist in achieving across institutions and communities participation, in the balance between top-down direction needed for technical infrastructure and grass-roots initiatives necessary to insure citizens' cohesion and finally in the balance between economic and civic motivations.

## 3 Legal and institutional norms in digital cities

In digital cities "government regulations extend laws with specific guidance to corporate and public actions" [2]. This legal aspect of norms is what we are concerned here as our context is digital cities, we will therefore only look at legal norms. In their introduction to normative multi-agent systems, Boella et al. give the following definition:

> A normative multi-agent system is a multi-agent 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." [4]

We first discuss the distinction among various kinds of norms, and then we discuss four issues in this definition, illustrated by examples in digital cities.

### 3.1 Kinds of norms

Several kinds of norms are usually distinguished in normative systems. Within the structure of normative multi-agent systems Boella et al distinguish "between regulative norms that describe obligations, prohibitions and permissions, and constitutive norms that regulate the creation of institutional facts as well as the modification of the normative system itself" [5]. A third kind of norms, procedural norms, can also be distinguished "procedural norms have long been considered a major component of political systems, particularly democratic systems" states Lawrence who further define procedural norms as "rules governing the way in which political decisions are made; they are not concerned with the content of any decision except one which alters decision-making procedures" [6].

**Constitutive norms:** Boella et al. note several aspects of constitutive norms, one is as intermediate concept exemplified by "X counts as a presiding official in a wedding ceremony", "this bit of paper counts as a five euro bill" and "this piece

of land counts as somebodys private property" [7]. Searle further explains that "the institutions of marriage, money, and promising are like the institutions of baseball and chess in that they are systems of such constitutive rules or conventions" [8]. In digital cities, examples are the marriage norms and voting int he sense that going through the procedure counts as a vote.

Boella et al further believe that "the role of constitutive rules is not limited to the creation of an activity and the construction of new abstract categories. Constitutive norms specify both the behavior of a system and the evolution of the system..." [5]. The dynamics of normative systems is here emphasized as in norms revision, certain actions count as adding new norms for instance amendments: "the normative system must specify how the normative system itself can be changed by introducing new regulative norms and new institutional categories, and specify by whom the changes can be done" [5]. Today "US government agencies are required to invite public comment on proposed rules" [2] This is done through the digital government interface and allow revisions to be traced.

Another aspect of constitutive norms is organizational and structural, that is, how roles define power and responsibilities and how various hierarchies structure groups and individuals. "Not only new norms are introduced by the agents playing a legislative role, but also that ordinary agents create new obligations, prohibitions and permissions concerning specific agents" [5].

**Regulative Norms:** As stated by Boella et al., "regulative norms are not categorical, but conditional: they specify all their applicability conditions"[5] furthermore "legal systems are often modelled using regulative norms, like obligations and permissions. However, a large part of the legal code does not contain prohibitions and permissions, but definitions for classifying the commonsense world under legal categories, like contract, money, property, marriage. Regulative norms can refer to this legal classification of reality" [7]. Regulative norms express constraints, obligations and prohibition for example in the Luxembourg e-city, citizens have to use pdf files and cannot use postscript formats to access the administration documents. Regulative norms also express permission, rights and powers, for example access rights or voting right if you are resident for more than 5 years or born in the city for Luxembourg. Another example is for creating a online library account on the Paris internet site, a parents' authorization is necessary if you are under 18 years old.

**Procedural norms:** Lawrence distinguishes two kinds of procedural norms "objective procedural norms are rules which describe how decisions are actually made in political systems; A systems objective procedural norms are a primary determinant of the content of political decisions in that they specify who actually makes decisions, who can try to influence decision makers, what political resources are legitimate and how resources may be used. Subjective procedural norms, on the other hand, are attitudes about the way in which decisions should be made" [6]. Procedural norms are instrumental for individuals working in a system. In digital cities, examples are back office procedures.

### 3.2 Explicit representation

The first property of norms in the definition of normative multi-agent systems is that norms are explicitly represented.

Norms are often given as requirements of computer systems and only implicitly represented. An example of implicit representation is a form in which you would be asked to state wether or not you keep a pet at home without mentioning to you the purpose of the information e.g. that if your answer is affirmative, you would be requested to pay a license fee or the fee would be directly taken from your bank account. Implicit representations are opaque to users and prevent governments to fulfill the democratic promise that transparency and explicit representations deliver. As users' needs for explanation and understanding of rules and regulations grow, representations have to become more explicit and personalized to their expectations. Similarly, governments'interests also reside in explicit representation to be addressed with the development of knowledge representation and reasoning mechanisms.

Current efforts are somewhat in-between implicit and explicit representation with tools for text representation and retrieval with more advanced ontologies, semantic links and search capabilities. "Businesses spend a lot of time complying with laws and regulations and worrying about what they don't know. According to a report by the Small Business Paperwork Relief Task Force, the Office of Management and Budget estimated that in fiscal year 2003, it took businesses and citizens approximately 8.2 billion hours and 320 billion dollars filling out paperwork and complying with government regulations" http://business.gov/. Indeed, this US governmental Business portal was launched in 2006 to help small businesses comply with Federal regulations, a need that was not being met by any other Federal government program.

In NYC, to renew a Driver License the stipulation is: "You cannot change your address during this transaction. You must have a completed form MV-619 (Eye Test Report) for this transaction. Read the requirements before you begin this transaction". To order a duplicate registration sticker for your car and get a receipt on-line, the stipulation is that (1) it replaces your damaged, lost or stolen registration items but not plates, (2) your duplicate will arrive by mail. Allow up to 2 weeks; most orders arrive in a week or less, (3) you should not order on-line if your address has changed, (4) if your registration is due to expire within 60 days, you may choose to renew now instead of ordering a duplicate and (5) you should be aware that you may be ticketed if your vehicle does not display a valid registration sticker. The quickest way to get your window sticker and registration receipt stub is to go to a local DMV office http://nyc.gov.

### 3.3 Norms can be violated

The second property in the definition of normative multiagent systems is that norms can be violated. This is also seen as an important condition for the use of deontic logic in computer science: "Importantly, the norms allow for the possibility that actual behaviour may at times deviate from the ideal, i.e. that violations

of obligations, or of agents rights, may occur", as observed by Jones and Carmo [9].

If norms cannot be violated then the norms are "regimented". For example, if there is a norm in access control that a service can only be accessed with some certificate, then this norm can be implemented in the system by ensuring that the service can only be accessed when the certificate is presented too. Regimented norms correspond to preventative control, in the sense that norm violations are prevented.

When norm violations are possible there is only detective control, in the sense that behavior must be monitored, and norm violations have to be detected and sanctioned. "social order requires social control, an incessant local (micro) activity of its units, aimed at restoring the regularities prescribed by norms. Thus, the agents attribute to the normative system, besides goals, also the ability to autonomously enforce the conformity of the agents to the norms, because a dynamic social order requires a continuous activity for ensuring that the normative systems goals are achieved. To achieve the normative goal the normative system forms the subgoals to consider as a violation the behavior not conform to it and to sanction violations" [7].

Norms can be violated because they are soft constraints. In digital cities, disincentives are often the mechanism used to prevent users from infringing their norms. In point (5) of the previous example, if you don't have a valid registration sticker on your windshield you may be ticketed. Similarly, most digital cities offer online appointments for pick up of large objects or appliances containing Freon, such as refrigerators, by their sanitary department. They clearly stipulate that illegal dumping is forbidden and will be fined, in Paris 183 euros.

When norm violations are possible, there are normative multiagent systems in which the violations can trigger new obligations, the so-called contrary-to-duty obligations. With contrary-to-duty obligations, there is not only a distinction between ideal and bad behavior, but there is also a distinction between various degrees of sub-ideal behavior.

Norm violations can lead to sanctions for compensation, or attempts to undo the violation (roll-back in database systems). When there is a sanction, the repair action can either be completely undone (as for example in business or economics) or we can remain in sub-ideal state (as in some kind of moral reasoning).

### 3.4 Where do they come from?

The definition of normative multiagent systems does not say where norms come from. There are many ways in which norms can be created. For example, in digital cities "government regulations extend laws with specific guidance to corporate and public actions" [2].

### 3.5 Can they change overtime?

A third issue in the definition of normative multiagent systems is that agents can modify norms.

In many electronic institutions the norms are fixed and cannot be changed within the system, but in many organizations there are roles defined within the system. The question is whether digital cities are a collection of electronic institutions, in other words, are manipulation and changes allowed within the system? In the US the site Regulations.gov provides a national forum for users to comment on existing and pending federal rules.

## 4 Definition of Conviviality

### 4.1 Introduction

The concept of conviviality often comes up in the context of web communities to describe sociable and forthcoming relations but it also arises in institutional contexts to generically denote the more specifically human qualities of communication: fun, easy going, friendly, cheerful, lively or polite. Looking at the term frequency on the Europe Information Society Thematic portal, if one enters today "convivial" in the search box, the result pages list 65 different documents. Indeed, in 1998 the European Community developed its strategy to promote conviviality as shared social value and selected it as research theme, one out of four, for its 5th Framework program for research (1998-2002): Societe de l'Information Conviviale, User-Friendly Information Society. It first called for large scale projects and programs that promoted user empowerment, human interactions, ambient intelligence and distributed services, for the elaboration of projects ranging from Content4All (2004-2006), to Humaine (2004-2008) to Companions (2006-2010) and programs such as the Convivio Net Consortium (2003-2005) developing convivial technologies, that is people-centered; All these initiatives seek to address the growing challenges in digital cities: need to support new interaction and communication paradigms, to bridge the increasing digital divides between social groups and remedy nascent social fragmentation and isolation by increasing social cohesion and community identity.

Generally speaking, a convivial place or group is one in which individuals are welcome and feel at ease [10] [11] [12], but definitions in literature spread from individual freedom realized in personal interdependence [13] to rational and cooperative behavior [14] to normative instrument when in the hands of power at play [15].

In this section we raise the following question: Which definition of conviviality can be used and operationalized for digital cities? By means of information and communication technologies, digital cities are virtual presences and extensions of our physical cities. Started in 1990 (see Figure 1) they divide into five broad categories [16]: Non-profit grass-root community initiatives from the early days, municipal information and communication networks now referred to as e-governments, commercial city-oriented web sites such as AOL Digital Cities and MSN Citisearch, virtual environments for virtual communities for example communities of interest and finally information and communication technology (ICT) experiments. Although initially an American phenomena, the European

Community quickly caught up in '93 with Telecities Network and in 2000 a 30-year plan encouraging member countries to build their own digital cities baed on common vision. Each step of the plan corresponds to a technological level: the current stage focuses on establishing systems interoperability, the following one on Intelligent City Systems (2009), then Ambient Intelligence (2013) and finally Smart Cities (2030). The principal objectives are to "transform and modernize local administrations in order to improve the level and quality of life of the population at both individual and community levels" [16], for example providing 24/7 access to services and content to reduce waiting lines and traffic congestion, or multilingual functionalities to reflect the linguistic diversity, facilitate inclusiveness process and reinforce social cohesion.

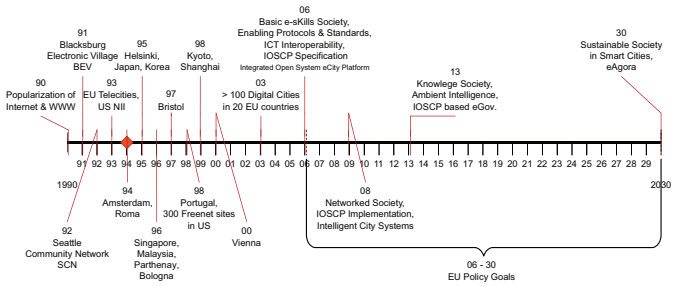



**Fig. 1.** Timeline: Conviviality and Ambient Intelligence

The question breaks down into the following questions: what kinds of notions of conviviality exist and how can these notions be used for digital cities. The methodology we follow for this section is a literature review in the areas of semiotics, philosophy, sociology, computer science, agent theory and human computer interaction. We then proceed with critical discussions. Our success criteria are the generality of our model and the evaluation and measure of conviviality. As stated by Sadek et al. [14], conviviality is "the essential and global characteristic of services () it emerges from the intelligence of the system and not from a set of local characteristics". Indeed, the local characteristics or criteria that determine conviviality "vary depending upon the application context and the types of users" [14]; consequently a list of criteria will not add up to a conviviality value. The critical factors are on one side the relation that binds the criteria together and on the other side, the way this relation is perceived by individuals. Furthermore, criteria are defined for specific contexts: security for banks, trust for relationships, flexibility and adaptability for web interface, scalability, efficiency and speed for systems, user density for given locations, group stability for communities of interest and so on.

In this paper we do not review the aspects of belief, desire and intentions of agent theory. Also out of scope are game theoretical approaches, the notion of equilibrium inherent to the temporal dimension of group behaviors and cost-

benefits analysis from economics theory. The layout of this section is as follows. In each section we first give an overview of the kinds of notions of conviviality existing in the field and then discuss how these notions can be used in digital cities. We start with socio-cognitive approaches, then we consider computer science, agent theory and multi-agent systems, and finally Human Computer Interaction.

### 4.2 Socio-Cognitive Approaches

**The role of conviviality in social systems** Conviviality describes a relation not only between the individuals of a group but also between groups. As power shifts between groups so does the way groups and individuals relate to each other. Conviviality then addresses the relations between groups of different characteristics, minority and majority groups and consequently the concepts of exclusion, the outsiders being kept away, and intrusion, the intruders forcing their way into the excluding group. This process reveals the dynamic aspect of conviviality and its temporal dimension and raise the questions of how is conviviality created, how it evolves and what makes it fail.

*Definitions of conviviality*

Looking at some definitions shows that the meaning of conviviality depends from the context of use (table 1). For example, a convivial technology applies to the relation human-computer and refers to how easy, efficient and intuitive it is for a human to use this technology; whereas in sociology it applies to the relation human-human and refers to the equally beneficial quality of the relation.

**Table 1.** Definitions of conviviality

| **Etymological and domain specific definitions of conviviality** |
|---|
| Origin: 15 th century "convival", from latin, convivere "to live together with, to eat together with". (French Academy Dictionnary) |
| Adj. Convivial: (of an atmosphere, society, relations or event) friendly and lively, (of a person) cheerfully sociable. (English Oxford Dictionary) |
| Technology: Quality pertaining to a software or hardware easy and pleasant to use and understand even for a beginner.(Adj.)User friendly, (Noun) Usability. By extension also reliable and efficient. (Grand Dictionnaire Terminologique) |
| Sociology: Set of positive relations between the people and the groups that form a society, with an emphasis on community life and equality rather than hierarchical functions. (Grand Dictionnaire Terminologique) |

A less common view of conviviality is when it becomes an instrument to exercise power and enforce one point of view over another. Conviviality is then experienced as a negative force by the loosing side. We summarized from different sources the positive and negative aspects of conviviality and as example present excerpts (table 2). Clearly, the positive aspects of conviviality emphasize the sharing of common grounds whereas the negative aspects emphasize division and coercive behaviors.

**Table 2.** Conviviality: Positive and negative aspects

| Positive aspects (enabler) |
|---|
| Share knowledge and skills |
| Deal with conflict |
| Inclusiveness |
| Equality |
| Trust |
| **Negative aspects (threat)** |
| Crush outsiders |
| Fragmentation |
| Totalitarism |
| Reductionism |
| Deception |

*Individuals vs. groups*

Being the first in 1958 to use conviviality in a scientific and philosophical context, Polanyi [17] describes it as synonymous with empathy "which alone can establish knowledge of other minds". Through empathy individuals identify with others, it provides a way to understand other individuals by experiencing their feelings, thoughts and attitudes thereby acquiring personal knowledge, yours and theirs. Polanyi further describes in 1974 [18] a community as convivial when it aims at sharing knowledge: members trust each others, share commitments and interests and make mutual efforts to create and preserve conviviality.

In his 1971 critical discourse on education, Deschooling Society [19], Illich defines a convivial learning experience as one based on role swapping: teacher role alternates with learner role and vice versa, thereby emphasizing the concept of reciprocity as key component to conviviality. It is however with Illich's 1973 publication of Tools for Conviviality [13] that the concept really acquires a new dimension as it is defined as "an intrinsic ethical value". For Illich, conviviality is synonymous with "individual freedom realized in personal interdependence", it is the foundation of a new society one that gives its members the means, referred to as tools, for achieving their personal goals: "A convivial society would be the result of social arrangements that guarantee for each member the most ample and free access to the tools of the community and limit this freedom only in favor of another member's equal freedom.".

However it is Putman and his colleagues who in the 80's take the concept of conviviality further as an enhancement to social capital and in 1988 refer to conviviality as a "condition for civil society" [20]. Putnam later argued then later in 2000 that in a civil society "communities are characterized by political equality, civic engagement, solidarity, trust, tolerance and strong associative life" [21]. These are the values found today in e-governments charters that aim at increasing social cohesion and inclusiveness, by putting citizens at the centre of technological change.

In 2004 Schechter, taking part to a semiotics symposium on conviviality, takes another look at the concept adding that "in a basic sense, conviviality is a social form of human interaction" [12]. She binds interaction to experience and recognizes the social dimension of conviviality as a way to reinforce group cohesion through the recognition of common values. "Thus the sharing of a certain kind of food and/or drink can be seen as a way to create and reinforce a societal group through a positive feeling of togetherness (being included in/or part of the group), on which the community's awareness of its identity is based." Schechter transforms the physical experience of conviviality into a learning and knowledge sharing experience. "To know is to understand in a certain manner that can be shared by others who form with you a community of understanding".

*From groups to institutions*

One cannot impose conviviality to a group claims Lomosits, it is a "consultative process that can only be recommended" [22] and reached through a consensus.

However, Hofkirchner explains that it is the "normative idea of unity-through-diversity that deserves attention when applying conviviality to the level of world society" [23]. The author then examines the unity-diversity relation. He first replaces the terms unity-diversity with identity-difference and then describes the four resulting scenarios: (1) "establish identity by eliminating difference at the cost of the differentiated side" yielding reductionism and universalism or (2) "of the undifferentiated side yielding unity without diversity", that is particularism, totalitarism and homogenization; (3) "establish difference by eliminating identity yielding diversity without unity", that is fragmentation and (4) "establish identity in line with difference yielding unity and diversity". The achievement of conviviality is in this integration of difference and differentiation of identity. Among other examples, it yields transculturalism.

Somov precises the normative aspect of conviviality in that it belongs to the regulation of human interrelations: "Conviviality (just like conflicts) is based on agreements or contradictions" [24].

According to Lamizet, conviviality was elaborated to describe both "institutional structures that facilitate social relations and technological processes that are easy to control and pleasurable to use" [25]. Conviviality emphasizes individual expression facilitated by personalized interface and customized content, while it also contributes to media standardization and the uniformization of representation systems.

*The darker side of conviviality*

"Conviviality is achieved for the majority, but only through a process by which non-conviviality is reinforced for the minority" states Ashby [26]. This aspect of conviviality rarely considered brings negative results along with the expected positive ones. Ashby further reveals the instrumentalization of conviviality to favor one group at the expense of another: "truth realities about minorities are built from the perspective of the majority via template token instances in which conflict is highlighted and resolution is achieved through minority assimilation to majority norms" [26].

Taylor Taylor [15] goes further: "Conviviality masks the power relationships and social structures that govern communities." The author explores the contradiction between institutions and conviviality raising 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". In Taylor's view 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."

**The use of conviviality for digital cities**

Users go to city web portals to fulfill needs for administration documents, official information, entertaintment and so on. When trying to accomplish their tasks, users have to struggle with a number of constraints and must continuously compromise. Constraints come from other users competing for resources or imposing their behaviors, from city administrations information and identity controls, or technical reasons such as system overloads or lack of functionality. Groupware, communityware and other research areas explore collaborative tools and systems however, there is to date no formal model for integrating notions like conviviality.

Conviviality affects coalition formation among humans by motivating individuals to associate with each other: it allows more efficient learning and reinforces social cohesion. Conviviality affects knowledge sharing and encourages cooperative behaviors, both constitutive values for digital cities. Moreover, conviviality contributes to reducing mis-coordinations that result from breakdowns in shared knowledge.

Some open questions are for example how to avoid reducing conviviality to one of its components and how to preserve its core value and meaning? A problem with formalization and implementation is that the concept of conviviality itself is inherently non-formal, when for instance you formalize it, the result may be a set of rules, norms, notions such as trust, reputation, bonus points, or other economic notions that miss the point. On analysis, conviviality may disappear and be reduced to other notions.

### 4.3 Computer Science Approaches

**The role of conviviality in Multi-agent systems (MAS)**

In multi-agent systems an agent is defined as "a computer system that is situated in some environment, and that is capable of autonomous action in this

environment in order to meet its design objectivesAgents are capable of flexible (reactive, proactive, social) behavior" [27]. This capability is crucial since it allows agents to cooperate, coordinate their actions and negotiate with each other.

*The use of conviviality for Intelligent Tutoring Systems*

The system proposed by Gomes et al. [28] provides a recommendation service of student tutors for computational learning environments. "Each agent pupil represents a pupil logged onto the system. One of the functions of the system is to be the client for an instant message service. Through its agent pupil, any pupil can communicate with other pupils in the system". Another function of agent pupils is to pass information, inferred by the agent or adjusted by pupils, on the pupils' affective states.

The authors' claim that "convivial social relationships are based on mutual acceptance through interaction" hence on reciprocity, here students helping each other. A utility function takes as input students' social profiles, computes students' affective states indicating their need of help and then recommends tutors. Remaining challenges are to define inputs for utility functions computing recommendations, presently random values, and automated inference of students in need. These critical tasks show the importance of further research in evaluation methods and measures for concepts such as mood, sociability and conviviality.

*The use of conviviality for Conversational Agent*

"All service offerings must integrate conviviality to the interaction between user and system as an essential preoccupation" [14]. To fulfill this goal, Sadek et al. define a convivial agent as rational and cooperative, consequently an interaction is convivial "if the agent presents, jointly and at all times, one or all of the following characteristics: Capacity for negotiation, contextual interpretation, flexibility of the entry language, flexibility of interaction, production of co-operative reactions and finally of adequate response forms." These communicative capacities and social intelligence based on emotional intelligence are crucial to enhance agents' ability to interact with users.

Building on this work Ochs et al. [29] distinguish felt emotions from expressed emotions noting that "a person may decide to express an emotion different from the one she actually felt because she has to follow some socio-cultural norms". We believe this direction to be very relevant to the evaluation of conviviality as it dissociates personal from social expression.

*The use of conviviality for reputation systems*

Reputation is defined as "the overall quality or character as seen or judged by people in general and the recognition by other people of some characteristic or ability" [30]. When Casare and Sichman state that "reputation is an indispensable condition for the social conviviality in human societies" [31] they emphasize this overall quality that both reputation and conviviality share. The authors' system insures that everyone is aware of anyone that complies or not to the rules of the group. The authors define a functional ontology of reputation for multi-agent systems whereby "roles are played by entities involved in reputative processes such as reputation evaluation and reputation propagation."

The authors' claim that "concepts of the legal world can be used to model the social world, through the extension of the concept of legal rule to social norm and the internalization of social mechanisms in the agent's mind, so far externalized in legal institutions". The agents actual behaviors are therefore compared to the social norms observed in their world. The process however presupposes an initial reputation profile of users that agents can then update in real time. Reputation acts as a communication tool ensuring complete social transparence throughout the system. The strict application of norms to reputation however may be difficult and suffer from rigidity. Of course, the same holds for conviviality.

**The role of norms in multi-agents systems and how it applies to conviviality**

The role of norms is increasingly getting attention specifically in multi-agents systems (MAS) where the most common view is that "norms are constraints on behavior via social laws" [4]. Boella et al. describe action models where "agents are goal directed and try to maximize their choice of means to obtain a goal". Moreover, it is assumed that an agent belongs to a group and must follow the norms like all members of that group. Similarly, using conviviality for a digital community maximizes the community benefits, for instance by reducing conflicts during communications allowing efficient interactions and cooperation. Indeed, conviviality, like politeness, uses the group rules to regulate members' interactions.

Boella et al. enunciate the three different functions of norms as follows: "we have norms that are of a constitutive nature, they define the agent's membership in a system of action, and the system of action at large. Another function of norms is regulation, describing what members of a social system must and must not do. Thirdly, norms may have a distributive function that is how rewards, costs and risks are to be divided among the social system's members" [4]. Similarly, conviviality reinforces social cohesion by reflecting its core values internally as well as externally. It also contributes to optimize performances within communities as well as between communities, improving coordination in the city. Finally, conviviality is enforced through mechanisms such as the expression of feelings: praise and encouragements for members who conform to the rules, anger and blame for the ones who do not.

### 4.4 Human Computer Interaction (HCI) Approaches

**Ambient spaces and playthings**

In her study of animated toys, Ackermann, looking at the relational qualities of playthings notes that beyond humanoid traits, it is an AniMate's manners of interaction that matter: "Beyond smarts, it is its conviviality. Beyond obedience or bossiness, it is an AniMate's relative autonomy and ability to share control" [10]. Ackermann emphasizes the exploration of partial and shared control as critical quality of conviviality. Merging digital and physical elements, she now experiments with ambient places, Piazzas, defined as convivial spaces and that function as transitional zones or third-space, between work and home.

**Pedagogy**

According to Sipitakiat, conviviality establishes a "dynamic equilibrium in the interplay between different modes of learning and teaching" [11]. In his research, based on constructionism and on Illich theories, the author asserts that "people learn with particular effectiveness when they are engaged in constructing personally meaningful artifacts (such as computer programs, animations, or robots)." The author stresses that conviviality encourages people to produce information and content rather than just consume it.

**User-friendly vs. convivial**

In HCI until recently, the terms user-friendly and convivial were often synonymous. However, the distinction increased as factors such as emotional experience and enjoyment were taken into consideration, user-friendliness referring now more to qualities such as ease-of-use, compliance to ergonomics standards and usability rules whereas the notion of conviviality finds new meaning with HCI developments in areas such as adaptive systems, augmented cognition, multi-modal interactions and ambient intelligence. Ackerman's research trying to define the qualities that make convivial spaces is just one of many examples relevant to the user of conviviality for digital cities.

### 4.5 Other Related Works

**Artificial sociable companions**

The Companions that Wilks envisions [32] are permanent software agents attached to single users. They act as intermediaries for all information sources that users cannot manage. For instance, Companions for seniors provide company to senior citizens who feel lonely, they act as technical task assistant to search the web for travels or keep track of events their owners forget. Conversely, Companions for juniors provide assistance with teaching, explanations-on-demand and advices.

**Mixed-Initiative Interaction** In a rather new area of research called mixed-initiative interaction "people and computers take initiatives to contribute to solving a problem, achieving a goal, or coming to a joint understanding" [33]. A critical element is how users focus their attention: "Attentional cues are central in decisions about when to initiate or to make an effective contribution to a conversation or project" [34]. Mixed-initiative research aims at developing software that filters appropriately incoming information to shield users from incoming disturbances such as emails and phone calls. The filtering of incoming information is achieved through measuring user's keystrokes and scrolling activities, recording the number of opened windows, analyzing content, checking events in calendars, location and time of day and so on.

**The cognitive dimension of conviviality** Research on sociable companions, information filtering, interruption and distraction clearly exemplifies the cognitive aspects of conviviality. It also suggests wide ranging uses for digital cities: from individual social assistant, to group communications, to regulation of emergent behaviors.

### 4.6 Our definition

We summarize by first noting that conviviality is usually considered a positive concept but that a darker side emerges when it becomes the instrument of power relations. Secondly, to the question we raised: Which definition of conviviality can be used and operationalized for digital cities? We answer with this two-fold definition of conviviality as

1. a condition for social interaction and
2. an instrument for the internal regulation of social systems.

Consequently, we see the most important uses of conviviality in digital cities as a mechanism to reinforce social cohesion as well as a tool to reduce mis-coordinations between individuals.

## 5 Use of Norms for Conviviality

In this section we reconsider the issues discussed in the context of legal and institutional norms for digital cities in the context of norms for conviviality.

### 5.1 Kinds of norms

The distinction between prohibitions and permissions (or rights) also occur for conviviality, as there are positive and negative aspects of this social concept.

Counts-as rules can be used to define kinds of conviviality.

### 5.2 Explicit vs. implicit representation of conviviality

Since most norms for conviviality are social norms, they are often not made explicit. Consider for example norms of being politically correct. Agents may appear to follow and embrace the beliefs of a group by fear of appearing different but without conviction, following a group without truly being part of it, and so on. Other implicit norms may refer to deception, non-transparent systematic controls, or to hide conflicts.

However, some explicit norms are relevant for conviviality. Explicit norms relevant for conviviality can refer to cooperation among agents, for example the protocol of communication, For example, http was successful thanks to its "conviviality", its limitations made it "popular" since they enhanced its ease of implementation. Ftp is successful too, for it compensates some of the http limitations, and both ftp and http are very cooperative. As another example, the digital city Issy les moulinaux [http://www.issy.com/] put all its accounting books online to all citizens.

### 5.3 Conviviality can be violated

There are many examples of violating conviviality. Ignoring cultural and social diversity is violating conviviality as it creates conviviality for a group at the expense of others. Another possible violation is being ignored when coming to ask advices to a city administrator. For example, the online Paris library assures members of a kind and pleasant service and proposes a free mediator service in case of difficulties dealing with city clerks. Yet another violation is to promote homogenization and enforce exclusion, or to crush outsiders, fragmentation, totalitarism, reductionism, deception and so on.

There are many possible solutions to conviviality violation, dealing with issues such as sharing knowledge and skills, deal with conflict, feeling of "togetherness", equality, trust and so on.

### 5.4 Do norms change over time?

Conviviality meant nothing to the IBM system developers in the 60's until the 90's. The IBM system command language "JCL" was deprived of any logic, coherence. Similarly the first IBM PCs did not even try to be accessible to the common users. Conviviality implied a form of democratisation of the use of computers. Eventually IBM turned to UNIX and Microsoft, with Windows 95 rediscovered that conviviality could be a strong marketing argument. Conviviality is then seen as a call to the user intelligence (whether he/she was a computer specialist or a simple end user), adaptability, curiosity, via a clear and communicable - i.e. teachable - protocol, aimed at providing an extensive control of systems which could be quite complex.

### 5.5 Where do norms come from?

The Socratic conviviality scheme, identifies two kinds of partners in the protocol: pseudo-passive ones (e.g. a guide, a master, a service provider, i.e., a reactive agent, but not one who takes the initiative) and active ones (e.g. those who are seeking assistance). The protocol consists in stimulating the intelligence and curiosity of the active partner, by questions of the pseudo-passive one. I.e. the active partner is considered an intelligent and autonomous agent : this active agent feels therefore he/she is a respected individual.

For instance, a tutoring system, asks as first question "may I help you?" and try to extract from a user answer - one could consider it as a Google query - possible tracks e.g. towards tutorials. Or that system could attempt to evaluate the user understanding level in some discipline, via a set of questions, and reading suggestions.

As a conversational agent, the Eliza system proved to be genuinely convivial, though extremely simple. It fostered the desire of its users to express and uncover themselves, by simply answering (often dumm but unexpected) questions on a terminal, as if the user was lying on a psychiatrist couch.

The Socratic protocol requires an open system, a system which needs permanent adaptations , since that protocol admits it will never be able to respond to all the user expectations, and the protocol has to have a "user criticism" or a "failure reports" section where users may report their frustration. The analysis of these user reactions is one of the major basis for the system enhancement and evolution.

Such a protocol could be quite "normative". This corresponds to the " conviviality " of an institutional relation, with its assumed teleology.

## 6 Conclusion

In this paper we consider contrast norms for conviviality with legal and institutional norms in digital cities. We consider the following issues. First, the kinds of norms typically distinguished in legal systems can be distinguished for norms of conviviality too. Second, norms for conviviality are often implicit, and we believe it is an important question when such norms should be made explicit. Third, the issue of violation of conviviality and ways to deal with it is of central concern in web communities like digital cities. Fourth, norms concerning conviviality should be able to change over time. Fifth, norms for conviviality can come from a wide variety of sources.

## References

1. Caire, P.: A critical discussion on the use of the notion of conviviality for digital cities. In: Proceedings of Web Communities 2007. (2007)
2. Lau, G.T., Law, K.H., Wiederhold, G.: Analyzing government regulations using structural and domain information. IEEE Computer **38** (2005) 70–76
3. den Besselaar, P.V., Melis, I., Beckers, D.: Digital cities: Organization, content, and use. In: Digital Cities. (2000) 18–32
4. Boella, G., van der Torre, L., Verhagen, H.: Introduction to normative multiagent systems. Computational & Mathematical Organization Theory **12** (2006) 71–79
5. Boella, G., van der Torre, L.W.N.: Regulative and constitutive norms in normative multiagent systems. In: KR. (2004) 255–266
6. Lawrence, D.G.: Procedural norms and tolerance: A reassessment. The American Political Science Review (1976)
7. Boella, G., van der Torre, L.W.N.: Constitutive norms in the design of normative multiagent systems. In: CLIMA VI. (2005) 303–319
8. Searle, J.R.: Speech Acts: An Essay in the Philosophy of Language. Cambridge University Press (1970)
9. Jones, A., Carmo, J. Handbook of Philosophical Logic. In: Deontic logic and contrary-to-duties. Kluwer Academic Publishers (2002) 265–344
10. Ackermann, E.K.: Playthings that do things: a young kid's "incredibles"! In: IDC '05: Proceeding of the 2005 conference on Interaction design and children, New York, NY, USA, ACM Press (2005) 1–8
11. Sipitakiat, A.: Digital technology for conviviality: Making the most of students' energy and imagination in learning environments. Master's thesis, MIT, Cambridge, MA,USA (2001)

12. Schechter, M.: Conviviality, gender and love stories: Plato's symposium and isak dinesen's (k. blixen's) babette's feast. Trans, Internet journal for cultural sciences **1** (2004)
13. Illich, I.: Tools for Conviviality. Marion Boyars Publishers (1974)
14. Sadek, M.D., Bretier, P., Panaget, E.: ARTIMIS: Natural dialogue meets rational agency. In: IJCAI (2). (1997) 1030–1035
15. Taylor, M.: Oh no it isn't: Audience participation and community identity. Trans, Internet journal for cultural sciences **1** (2004)
16. Ishida, T.: Understanding digital cities. In: Digital Cities. (2000) 7–17
17. Polanyi, M.: Personal Knowledge: Towards a Post-Critical Philosophy. Routledge & Kegan Paul Ltd, London (1958)
18. Polanyi, M.: Personal Knowledge : Towards a Post-Critical Philosophy. University Of Chicago Press (1974)
19. Illich, I.: Deschooling Society. Marion Boyars Publishers, Ltd. (1971)
20. Putnam, R.D.: Diplomacy and domestic politics: The logic of two-level games. International Organization **42** (1988) 427–460
21. Putnam, R.D.: Bowling alone: the collapse and revival of american community. In: CSCW. (2000) 357
22. Lomosits, H.: Future is not a tense. Trans, Internet journal for cultural sciences **1** (2004)
23. Hofkirchner, W.: Unity through diversity.dialectics - systems thinking - semiotics. Trans, Internet journal for cultural sciences **1** (2004)
24. Somov, G.Y.: Conviviality problems in the structure of semiotic objects. Trans, Internet journal for cultural sciences **1** (2004)
25. Lamizet, B.: Culture - commonness of the common? Trans, Internet journal for cultural sciences **1** (2004)
26. Ashby, W.: Unmasking narrative: A semiotic perspective on the conviviality/non-conviviality dichotomy in storytelling about the german other. Trans, Internet journal for cultural sciences **1** (2004)
27. Wooldridge, M.: An introduction to multi-agent systems. J. Artificial Societies and Social Simulation **7** (2004)
28. (Gomes, E.R., Boff, E., Vicari, R.M.)
29. Ochs, M., Niewiadomski, R., Pelachaud, C., Sadek, D.: Intelligent expressions of emotions. In: ACII. (2005) 707–714
30. Merriam-Webster, I.: Merriam Webster OnLine Dictionary. Merriam-Webster (2006)
31. Casare, S., Sichman, J.: Towards a functional ontology of reputation. In: AAMAS '05: Proceedings of the fourth international joint conference on Autonomous agents and multiagent systems, New York, NY, USA, ACM Press (2005) 505–511
32. Wilks, Y.: Artificial companions. In: MLMI. (2004) 36–45
33. Horvitz, E., Koch, P., Apacible, J.: Busybody: creating and fielding personalized models of the cost of interruption. In: CSCW. (2004) 507–510
34. Horvitz, E., Kadie, C.M., Paek, T., Hovel, D.: Models of attention in computing and communication: from principles to applications. Commun. ACM **46** (2003) 52–59

# Agents, Norms and Forest Cleaning

Jan Odelstad

Department of Mathematics, Natural and Computer Sciences, University of Gävle,
SE-801 76 Gävle, Sweden,
jod@hig.se

**Abstract.** The automation of forest cleaning presupposes principles for choosing those trees that ought to be taken away and those that shall be left standing. In this paper, which is a report on a work in progress, 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 that the agent system can evaluate the efficiency of the normative system and the utility function and, furthermore, suggest improvements of them. Earlier works on norms and norm-regulation of agent systems that the author has been involved in are used to elucidate the problem area discussed in the paper.

## 1 Introduction

Within economic theory the consumer's behaviour has traditionally been described as determined by a utility function. During the latest three decades there has been a growing interest among researchers in how norms (for example rules of law) give restrictions on the behaviour induced by the utility function.[1] The behaviour of the consumers or other economic agents, according to this model, is the result of 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.

In this preliminary report on a work in progress, I will discuss some aspects of how norms can be used to regulate the behaviour of multiagent-systems. I will do this at least partially with a concrete problem area in view, namely the automation of forest management treatments, especially the cleaning of young forest stands. The automation of forest cleaning presupposes principles for choosing, in a state of incomplete information, those trees that ought to be taken away

[1] Cf. for example [4] p. 518, where the so called Coase theorem is discussed.

Dagstuhl Seminar Proceedings 07122
Normative Multi-agent Systems
http://drops.dagstuhl.de/opus/volltexte/2007/917

and those that shall be left standing. I will discuss the possibility of formulating those principles as a kind of norms and in that way partially regulate the cleaning process by a normative system.

Norm-regulation of agents presupposes a precise and significant representation of norms and normative systems. 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 a permissible state it prefers the most.

In [14] 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 developed by Lindahl and myself. One of the results in [14] was a scheme for how normative positions will restrict the set of actions that the agents are permitted to choose from. The possibility of using the proposed theory in the construction of abstract architectures for multiagent-systems was discussed. The method used for describing abstract architectures was based on the definition of set-theoretical predicates.

Thee is a need for revisions and developments of [14] in different aspects. The algebraic representation of normative systems has been further developed since [14] was published, for example by the use of intermediaries, especially open intermediaries. The application of the Kanger-Lindahl typology of normative positions can be a more complex task than indicated in [14], for example two-agent types of normative positions can be used in addition to just one-agent-types. Furthermore, the characterization of a norm-regulated multiagent-system can be made more flexible. The intended extension and revision of [14] will be tested by an application to the decision-making problem of an idealized forest cleaning agent.

## 2 Automation of forest cleaning[2]

### 2.1 Cleaning *in silico*

In 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 a lot of difficult problems must be solved, for example how the robots will perceive the environment and how they will transport itself. 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 of which trees shall be

[2] This section is based on [2].

taken away (removed) and which will be left standing. Such "remove-decisions" must be made on-line with information only of the robot's neighbourhood and about that part of the stand already cleared. The treatment cannot be evaluated until the actual stand is completely cleared. To test and evaluate principles for remove-decisions by real-world-experiments (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 [1] a platform for simulation of young forest stands is presented. Given field data of a special type of young forest, for example a 10 years old somewhat damp spruce forest in the middle of Sweden on 200 m above sea level, 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 goal with 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 bases 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 formulate the principles as a normative system supplemented by a utility function, i.e. if the automation of forest cleaning could be modelled as an *Agent oeconomicus norma.*

Forest cleaning is a kind of activity and to elucidate some formal aspects of norm-regulation of forest cleaning, the next subsections are devoted to some general remarks on the structure of an activity and its evaluation.

### 2.2 The formal structure of an activity

An activity is based on actions performed by one or more agents. Accordingly, one can distinguish between one-agent-activities and multi-agent-activities. The execution of a specific instance of an activity involves the performance of a number of actions, often appropriately represented as a sequence of actions. Let $V$ be an activity and $c$ an instance of $V$. The execution of the instance $c$ of the activity $V$ starts from an input and results in an output. Let $c^i$ be the input of $c$ and $c^o$ the output of $c$. The input of $c$ is often an initial state while the output of $c$ is often a set of possible final states. If the output of an instance of the activity $V$ is always singleton, then the execution of $V$ is deterministic. If $c$ is an instance of a deterministic activity $V$ and $c^o = \{s\}$ we will often, for the sake of simplicity, say that $c^o = s$, i.e. we identify the output with the element in the singleton set.

Suppose that $V$ is an activity. There may be different ways of (or different procedures for) executing instances of the activity $V$ and each way or procedure

is represented by a function $F$ such that $F(c^i) = c^o$. If $F$ and $G$ are different ways of executing $V$ we say that $F$ and $G$ are extensionally equivalent, which is denoted $F =_e G$, if $F(c^i) = G(c^i)$ for all instances $c$ of $V$.

There are different modes of definition of $F$ and depending on the mode of definition there are different kinds of questions to ask. We will here give three examples.

(I) $F$ is determined as the agent $\omega$'s way of executing the activity $V$. This presupposes that the agent is sufficiently "reliable" such that one can suppose that the agent has a procedure for executing the activity $V$. In this case one can be interested in
   (i) making explicit the rules that $\omega$ uses, such that every competent agent can determine the value of $F$ for an argument $c^i$ by applying the rules, or
   (ii) determining a computational function $G$ which is extensionally equivalent to $F$.

(II) $F$ is defined by a system of rules such that an agent system $A$ which complies with the rules "computes" $F$. In this case one may want to characterize $F$ as a computationally more effective function that is not necessarily agent-based.

(III) $F$ is defined as a computational function, in which case it can be of interest to determine a system of rules such that if an agent follows the rule the result will be $F(c^i)$ for all instances $c$ of $V$.

### 2.3 Forest cleaning as a kind of activity

It does not seem to be adequate to consider 'forest cleaning' as an activity - it is too broad a concept and is more properly classified as a kind of activity. However, the cleaning of a special kind or type of forest for a certain aim or purpose is an activity. An instance of a cleaning activity is the cleaning (specified in an appropriate way) of a stand. Input of the instance is the stand before cleaning and the output is (a) the stand after cleaning if the activity is deterministic and (b) the set of possible results of cleaning the stand if the cleaning activity is indeterministic. A cleaning activity is often executed by one agent alone and cleaning is therefore in many contexts a one-agent-activity.

Let $\mathfrak{R}$ denote the activity 'cleaning of the kind $\kappa$ of forests for the purpose $\pi$'. $\kappa$ can, for example, be a 'mixed forest' or a 'deciduous forest'. $\pi$ can be related to the intended use of the forest in the future. An instance $c$ of this activity is the cleaning of a certain stand $p$. The input of the instance, i.e. $c^i$, is the stand $p$ before it is cleaned and we denote it $p_0$. This means that $c^i = p_0$. The output of $c$ is the stand $p$ after the cleaning has been done. Agents executing $\mathfrak{R}$ are today human beings. A human cleaner who is going to execute the instance $c$ does not need a description of $p_0$, but an artificial cleaner needs a description or representation of the input of $c$. Such a representation may consist, for example, of a description of each tree in the stand, the position of the trees, the topography of the land, the quality of the soil, the existence of obstacles (for example large stones and ditches) and the climate (micro as well as large scale) of the stand.

For testing different ways of executing cleaning, it is convenient to use a platform for simulations of young forest stands and such a platform is described in [7] and [1]. The platform is used for two different tasks, (a) to create replicate forest stands and (b) to perform cleaning on field data or on simulated data. The objective behind the first task is to create replicates of forest stands that belong to certain forest types. The basic data for a replicate simulation can be field data or simulated data. Data of a given forest type can be used for constructing a decision tree structure and probabilities are calculated for different parameter values of a tree, such as diameter, eventual damage and species. The decision tree structure can then be used for creating replicate forest stands that are different but represent the same forest type as the stand form which the data were collected. A user can also apply her own principles of forest stand parameters and thus decide more or less strictly how the resulting forest is going to be.

In [17], Vestlund et.al. describe a way of cleaning young forest stands as an on-line algorithm expressed in an informal language. The algorithm is at least partially based on interviews with professional cleaners. In [17] there is also an implementation of the algorithm in a programming language and cleaning *in silico* of forest stands represented by computer files are executed. In [2] results of cleaning simulated stands using principles based on the work by Vestlund et.al. are presented.

### 2.4 Evaluating activity procedures

Different procedures for executing an activity $V$ can be more or less good or appropriate. It is therefore important in many contexts to evaluate different ways of executing activities. Suppose that $F$ and $G$ are different ways of executing $V$. It is important to note the difference between how good $F$ is as a way of executing a given instance of $V$ and how good $F$ is a way of executing $V$ in general. It is thus important to distinguish between

1. $F(c^i) \succ^V G(c^i)$, i.e. the execution of the instance $c$ of $V$ by $F$ is better than by $G$.
2. $F \succ_V G$, i.e. $F$ is quite generally a better way of executing $V$ than $G$.

In certain cases there exists a measure $u_V$ of how good different procedures for executing an instance $c$ of the activity $V$ is and a measure $U_V$ of how good different procedures for the activity $V$ is generally. It is reasonable to suppose that the following holds:

(a) $u^V(F(c^i)) > u^V(G(c^i))$ iff $F(c^i) >^V G(c^i)$
(b) $u_V(F) > u_V(G)$ iff $F >_V G$.

If for all instances $c$ of an activity $V$ it holds that $F(c^i) >^V G(c^i)$, then it seems uncontroversial to conclude that $F >_V G$. But if for some instances $c$ it holds that $F(c^i) >^V G(c^i)$while for other instances $c$ it holds that $G(c^i) >^V F(c^i)$ then it is more difficult, in many cases impossible, to decide if $F >_V G$ or not.

The binary relations $>^V$ and $>_V$ and the measures $u^V$ and $u_V$ concern how good different execution procedures are (for given instances or generally) "all things considered". These relations and measures can be difficult to establish since appropriateness ("goodness"), all things considered, is in many contexts a very complex attribute. However, appropriateness with respect to a certain aspect (attribute) $\alpha$ may be easier to ascertain. Let

- $F(c^i) >_\alpha^V G(c^i)$ denote that $F$ is a better way with respect to $\alpha$ than $G$ to execute the instance $c$ of $V$; if the activity is cleaning then $\alpha$ can be for example the number of trees per unit area or the average distance between the trees.
- $F >_{V,\alpha} G$, denote that $F$ is a better way with respect to $\alpha$ than $G$ to execute $V$.
- $u_\alpha^V(F(c^i))$ is a measure of how good $F$ is with respect to $\alpha$ as a way of executing the instance $c$ of the activity $V$.
- $u_{V,\alpha}(F)$ is a measure of how good $F$ is generally with respect to $\alpha$ as a way of executing the activity $V$.

$u_\alpha^V$ and $u_{V,\alpha}$ can be considered as a kind of utility function. Utility functions are largely used in economics. In addition, utility functions play an important role in the study of intelligent agents within the discipline of artificial intelligence, as the following quotation from [16] p. 52 emphasizes: "... when there are conflicting goals, only some of which can be achieved ..., the utility function specifies the appropriate tradeoffs. ... when there are several goals that the agent can aim for, none of which can be achieved with certainty, utility provides a way in which the likelihood of success can be weighed up against the importance of the goals."

Suppose that different procedures for executing $V$ is evaluated with respect to the attributes $\alpha_1,\ldots,\alpha_k$. Then

$$u^V\left(F\left(c^i\right)\right) = f\left(u_{\alpha_1}^V(F(c^i),...,u_{\alpha_k}^V(F(c^i))\right).$$

The form of the function $f$ can differ considerably in different contexts. But it is often a desideratum that $f$ has a simple form, for example being additive or multiplicative.

For many aspects $\alpha$ of importance for how good $F$ is as a way of executing $V$, but certainly not for all aspects, it seems reasonable that the following holds:

- $F(c^i) >_\alpha^V G(c^i) \Rightarrow F(c^i) >^V G(c^i)$ ceteris paribus (everything else held constant)
- $F >_{V,\alpha} G \Rightarrow F >_V G$ ceteris paribus (everything else held constant).

It is of course desirable to have a complete method for evaluating different ways of executing an activity but when this is not possible the concepts introduced above can still make a partial evaluation possible, and this may in certain situations be of great interest.

Suppose that the relation $>_V$ is defined for the activity $V$. Then an optimal way of executing $V$ is a procedure $F$ that is maximal with respect to $>_V$. It

is usually desirable to choose an optimal way of executing $V$, but it can be difficult to find such a way if the number of different ways to execute $V$ is large. The search for an optimal way of executing $V$ may then be replaced by a more restricted goal, viz. the search for an acceptable way, i.e. a way that is good enough (sufficiently acceptable).

## 3 Norm-regulation of cleaning agents[3]

In [14], an abstract architecture for idealized multi-agent systems, whose behaviour is regulated by normative systems, is developed. The idea behind the architecture is roughly the following:

"When it is agent $\omega$'s turn to move it chooses an act out of a set of feasible alternatives and the result will be that the system enters a new state; which state depends on the actual state of the system when the act is performed. The agent's choice is determined partially by the preference ordering of the possible states and partially by the deontic structure: the agent chooses that act which leads to the best outcome of all permissible actions. If an action is permissible or not depends on whether the result of performing the action leads to a state which satisfies a condition which is forbidden according to the normative system regulating the multi-agent system." ([14] p. 152).

The architecture is articulated as a definition of a deontic action-logic based multiagent-system. Since 'MAS' is the standard abbreviation of 'multiagent-system', we call the kind of system discussed here 'DALMAS'.

In this section the possibility to use DALMAS as the architecture for a cleaning agent is outlined. 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 the 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.

The formal definition of a DALMAS is summarized in the next section. In this section is given an outline of how the definition can be applied in the context of cleaning. Let us consider the cleaning of a stand $p$.

The stand $p$ is divided into $n$ different areas. A state 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. We let each area be denoted by a unique number between 1 and $n$. Let $S_i$ be the $i$:th state. Let $C_i$ be 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 $U_i$ $n - i$ 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 $k \in U_i \backslash \{j\}$.

[3] This section is based on [1].

The norm of a cleaning DALMAS is expressed in a conditional sentence of the following form, where DEOP is a deontic operator, for example 'it shall be the case that', 'it may be the case that ' or 'it may not be the case that':

Given that the actual state of the system has the property $P$, then DEOP(the next state of the system has the property $Q$).
Let us give a few examples of sentences that have the right form for being norms (which of course does not imply that they are norms):

(a) If there is only one undamaged tree in the area to be cleaned with 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 diameter within the desirable range, then a damaged tree with 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 diameter within the desirable range and with distances to other undamaged trees larger than 0.5m, then $t$ may not be saved.

The antecedent of a norm is thus a descriptive sentence and the consequent is a deontic sentence. In the definition of DALMAS, the Kanger-Lindahl theory of normative positions is used to obtain a logically powerful framework for expressing deontic sentences.

In many situations, the norms for a DALMAS do not determine the action to be taken in each state, but utility considerations are also necessary. If we have the relation $>_{\mathfrak{R}}$ at hand, where $\mathfrak{R}$ denotes the activity 'cleaning of the kind $\kappa$ of forests for the purpose $\pi$', then we can search for the optimal way of cleaning the actual area, given the condition that the way of cleaning satisfies the given norms.

## 4 The definition of a Dalmas

The aim of [14] was to present a theory 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. The theory can be summarized as follows. The norms for a MAS are regarded as belonging to a normative system and such a system is represented algebraically as a Boolean joining system containing a Boolean quasi-ordering of grounds and a Boolean quasi-ordering of consequences. The norms are joinings from the Boolean quasi-ordering of grounds to the Boolean quasi-ordering of consequences, and the specific normative content of a normative system is given by the set of minimal norms. The consequences are expressed using operators on conditions corresponding to the Kanger-Lindahl-types of one-agent-positions. An important step in the theory construction was the specification under what circumstances the sentence $T_i d(\omega_1, ..., \omega_\nu, \omega; \omega, s)$ implies that an action $a$ is prohibited for the agent $\omega$ in the state $s$. (See section 6.) An action $a$ was regarded as a function, which is the usual way of representing an action in decision theory, and $d$ is a $\nu$-ary condition on agents, true or

false in the situation $s$. An abstract architecture based on the theory of norm-regulation of behaviour was defined, and a MAS having this architecture was called a norm-regulated DALMAS.

DALMAS is a global clock (synchronous update), global state, global dynamics system. It can be viewed as a simplification constructed for conceptual and computer simulation purposes. In particular, DALMAS can be used as a model system for studying the interplay between preferences and norms in MAS architectures.

A DALMAS is an ordered 7-tuple $\langle \Omega, S, A, \mathcal{A}, \Delta, \Pi, \Gamma \rangle$ containing

- an agent set $\Omega$ ($\omega, \varkappa, \omega_1, ...$ elements in $\Omega$),
- a state or phase space $S$ ($r, s, s_1, ...$ elements in $S$),
- an action set $A$ such that for all $a \in A$, $a : \Omega \times S \longrightarrow S$ such that $a(\omega, r) = s$ means that if the agent $\omega$ performs the act $a$ in state $r$, then the result will be state $s$ ($a, b, a_1, ...$ elements in $A$),
- a function $\mathcal{A} : \Omega \times S \longrightarrow \wp(A)$ where $\wp(A)$ is the power set of $A$; $\mathcal{A}(\omega, s)$ is the set of acts accessible (feasible) for agent $\omega$ in state $s$,
- a deontic structure-operator $\Delta : \Omega \times S \longrightarrow \mathcal{D}$ where $\mathcal{D}$ is a set of deontic structures of the same type with subsets of $A$ as domains and $\Delta(\omega, s)$ is $\omega$'s deontic structure on $\mathcal{A}(\omega, s)$ in state $s$,
- a preference structure-operator $\Pi : \Omega \times S \longrightarrow \mathcal{P}$ where $\mathcal{P}$ is a set of preference structures of the same type with subsets of $A$ as domains and $\Pi(\omega, s)$ is $\omega$'s preference structure on $\mathcal{A}(\omega, s)$ in state $s$,
- a choice-set function $\Gamma : \Omega \times S \longrightarrow \wp(A)$ where $\Gamma(\omega, s)$ is the set of actions for $\omega$ 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 $\omega$ 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 $\Omega \times S$.

In a DALMAS, all the agents have the same initial set of actions. The set of actions to choose from (the choice-set) in a situation $\langle \omega, s \rangle$ is determined by the agent's deontic structure $\Delta(\omega, s)$ and preference structure $\Pi(\omega, s)$. If $\Gamma(\omega, s)$ consists of one action, then this action applied in the situation $\langle \omega, s \rangle$, i.e. $[\Gamma(\omega, s)]\,(\omega, s)$, is the resulting state when $\omega$ acts in state $s$.

A *simple* DALMAS is a DALMAS containing the following simple versions of $\Delta$, $\Pi$ and $\Gamma$.

1. $\Delta(\omega, s) \subseteq \mathcal{A}(\omega, s)$ and $\Delta(\omega, s)$ is the set of permissible actions for $\omega$ in the state $s$,
2. $\Pi(\omega, s) = \langle \mathcal{A}(\omega, s), \succsim \rangle$ where $\succsim$ is a weak ordering
3. $\Gamma(\omega, s) = \{x \in \Delta(\omega, s) : \text{ for all } y \in \Delta(\omega, s),\ x \succsim y\}$.

Hence, in a simple DALMAS the choice-set consists of the best actions which are permissible. Among the elements in $A$ there can be a pass action, which

means the agent does nothing. If we combine the existence of such an action with very short clock cycles, we obtain systems with close to asynchronous behaviour.

A DALMAS is not deterministic, since it does not determine in which order the agents are going to move, and the choice-set may contain more than one action in every situation. Let us therefore make the following definition.

**Definition 1.** *A* deterministic DALMAS *is an ordered 9-tuple* $\langle \Omega, A, S, \mathcal{A}, \Delta, \Pi, \Gamma, \tau, \gamma \rangle$ *such that* $\langle \Omega, A, S, \mathcal{A}, \Delta, \Pi, \Gamma \rangle$ *is a* DALMAS *and* $\tau : \Omega \longrightarrow \Omega$, $\gamma : \wp(A) \longrightarrow A$.

The intended interpretation is the following:

- $\tau : \Omega \longrightarrow \Omega$ is a turn-operator such that $\tau(\omega) = \varkappa$ means that it is $\varkappa$'s turn after $\omega$; $\tau$ determines a simple agent priority.
- $\gamma : \wp(A) \longrightarrow A$ is a tie-breaking function, determining which of several permissible and equally preferred actions to choose.

An important tool in the present study is the characterization of abstract architectures by the definitions of set-theoretical predicates. Among the abstract architectures defined in this way, the most important one is a norm-regulated DALMAS. This is just the first step towards a theory of architectures for MAS that restricts the behaviour of the multiagent-system using norms. The theory can be developed by defining a number of set-theoretical predicates that are specifications of the predicate DALMAS, and we can obtain a hierarchy of predicates with DALMAS as its root.

## 5 The algebraic representations of norms and normative systems

The method used for representing norms in an architecture for norm-regulated MAS can be of importance for the effectiveness of the architecture. Let me mention a few examples of what can be regarded as desiderata for a norm-representation method.

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 a series of papers (among which are [15], [9], [10], [11] and [12]) Lars Lindahl and I have developed an algebraic approach to the study of normative systems. One of our main tools in this endeavour is the theory of a Boolean quasi-ordering, which is an extension of the theory of Boolean algebras. A norm is regarded as consisting of two objects, a ground and a consequence standing in a 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. A normative system $\mathcal{S}$ can therefore be represented as a Boolean joining system $\langle \mathcal{B}_1, \mathcal{B}_2, J \rangle$ where $\mathcal{B}_1$ is a Boolean quasi-ordering of ground-conditions, $\mathcal{B}_2$ a Boolean quasi-ordering of consequence conditions and the set $J$ of norms where $J \subseteq B_1 \times B_2$. One can define a qu asi-ordering $\trianglelefteq$ expressing how narrow norms are and determine the set of minimal elements of $J$, $\min J$, with respect to $\trianglelefteq$. 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 $C$ for a base of minimal elements of $J$. In many contexts the elements in $C$ can be represented by intermediaries (intermediate concepts). Intermediaries are determined by the condition that constitute its ground and the condition that constitutes its consequences.

Within the algebraic representation of norms the consequences are normative conditions. In [14] simple normative conditions were constructed by letting a "normative position operator" $T_i$, $1 \leq i \leq 7$, operate on descriptive conditions. Compound conditions are Boolean combinations of simple conditions. The operators $T_1, ..., T_7$ are applications of Lindahl's one-agent-types of normative positions, which are briefly described in the next section.

## 6 The Kanger-Lindahl typology of normative positions

Based on the work by Stig Kanger ([5], [6]), Lars Lindahl developed three systems of types of normative positions, see [8]. The simplest one is the system of one-agent types of normative position, and in [14] we made only use of this system. This kind of types is constructed in the following way. Let $\pm\alpha$ stand for either of $\alpha$ 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.[4] 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 [8] p. 92).

[4] The notion of 'maxiconjunction' was introduced in Makinson (1986) p. 405f.

$\mathbf{T}_1(x,q)$ : MayDo$(x,q)$ & MayPass$(x,q)$ & MayDo$(x,\neg q)$.
$\mathbf{T}_2(x,q)$ : MayDo$(x,q)$ & MayPass$(x,q)$ & $\neg$MayDo$(x,\neg q)$
$\mathbf{T}_3(x,q)$ : MayDo$(x,q)$ & $\neg$MayPass$(x,q)$ & MayDo$(x,\neg q)$.
$\mathbf{T}_4(x,q)$ : $\neg$MayDo$(x,q)$ & MayPass$(x,q)$ & MayDo$(x,\neg q)$.
$\mathbf{T}_5(x,q)$ : MayDo$(x,q)$ & $\neg$MayPass$(x,q)$ & $\neg$MayDo$(x,\neg q)$.
$\mathbf{T}_6(x,q)$ : $\neg$MayDo$(x,q)$ & MayPass$(x,q)$ & $\neg$MayDo$(x,\neg q)$
$\mathbf{T}_7(x,q)$ : $\neg$MayDo$(x,q)$ & $\neg$MayPass$(x,q)$ & MayDo$(x,\neg q)$.

$\mathbf{T}_1$,...,$\mathbf{T}_7$ are called the types of one-agent positions. Given the underlying logic, the one-agent types are mutually disjoint and their union is exhaustive. Note that $\neg$MayDo $(x,q)$ & $\neg$ MayPass $(x,q)$ & $\neg$MayDo$(x,\neg q)$ is logically false, according to the logic of Shall and May.

The ground of a norm is usually a descriptive condition, while the consequence is a deontic condition. In [10] we use the one-agent-types in the Kanger-Lindahl theory of normative positions as operators on descriptive conditions to get deontic conditions. As a simple example, suppose that $r$ is a unary condition. Then $T_i r$ (with $1 \leq i \leq 7$) is the binary condition such that

$$T_i r(y,x) \text{ iff } \mathbf{T}_i(x, r(y)),$$

where $\mathbf{T}_i(x,r(y))$ is the $i$:th formula of one-agent normative positions. Note that for example $\mathbf{T}_3(x,r(y))$ means

$$\text{MayDo}(x,r(y)) \;\&\; \neg\text{MayPass}(x,r(y)) \;\&\; \text{MayDo}(x,\neg r(y)).$$

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 $\mathbf{T}_i$.

## 7 Normative positions regulating actions

The idea behind a norm-regulated DALMAS is that the deontic structure operator $\Delta$ is defined in terms of a normative system $\mathcal{S}$ in the sense that what is permissible to do in a situation is determined by $\mathcal{S}$. This idea can be explicated in the following way. Let

$T_i d(\omega_1,...,\omega_\nu,\omega;\omega,s)$ mean that in the situation where it is $\omega$'s turn to draw and the state of the system is $s$, $\omega$ has the normative position of type $T_i$ with regard to the state of affairs $d(\omega_1,...,\omega_\nu)$.

Prohibited$_{\omega,s}(a)$ mean that in the situation where it is $\omega$'s turn to draw and the state of the system is $s$, $\omega$ is prohibited to execute the act $a$.

The following seven principles establish connections between the condition $T_i d$ and the predicate Prohibited (see [14] 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 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 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 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 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 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 Prohibited$_{\omega,s}(a)$.

These principles can be used to define the deontic structure-operator $\Delta$. One possibility is to let $\Delta(\omega, s)$ be the set of feasible acts $a$ that are not eliminated as Prohibited$_{\omega,s}(a)$ according to the rules (1)-(7) above, where

Prohibited$_{\omega,s}(a)$ is equivalent to $\neg$Permissible$_{\omega,s}(a)$.

Hence,

$\Delta(\omega, s) = \{\text{Permissible}_{\omega,s}(a) : a \in A\}$.

Note that in the outset all feasible acts are permissible, i.e. for all $a \in A$ Permissible$_{\omega,s}(a)$. The basic idea is that we eliminate elements from the set of permissible acts for $\omega$ in $s$ using the norms and sentences expressing what holds for the agents with respect to grounds in the norms. To facilitate the presentation it is convenient to introduce the following six operators on state-conditions:

$E_2^a d(\omega_1, ..., \omega_\nu, \omega; \omega, s)$ iff $[d(\omega_1, ..., \omega_\nu; s)$ and $\neg d(\omega_1, ..., \omega_\nu; a(\omega, s))]$
$E_3^a d(\omega_1, ..., \omega_\nu, \omega; \omega, s)$ iff $[d(\omega_1, ..., \omega_\nu; s)$ iff $d(\omega_1, ..., \omega_\nu; a(\omega, s))]$
$E_4^a d(\omega_1, ..., \omega_\nu, \omega; \omega, s)$ iff $[\neg d(\omega_1, ..., \omega_\nu; s)$ and $d(\omega_1, ..., \omega_\nu; a(\omega, s))]$
$E_5^a d(\omega_1, ..., \omega_\nu, \omega; \omega, s)$ iff $[\neg d(\omega_1, ..., \omega_\nu; a(\omega, s))]$
$E_6^a d(\omega_1, ..., \omega_\nu, \omega; \omega, s)$ iff not $[d(\omega_1, ..., \omega_\nu; s)$ iff $d(\omega_1, ..., \omega_\nu; a(\omega, s)]$
$E_7^a d(\omega_1, ..., \omega_\nu, \omega; \omega, s)$ iff $d(\omega_1, ..., \omega_\nu; a(\omega, s))$.

Note that for all $i$, $2 \leq i \leq 7$, $(T_i d \wedge E_i^a d)(\omega_1, ..., \omega_\nu, \omega; \omega, s)$ implies that Prohibited$_{\omega,s}(a)$

In [3] an implementation in Prolog of the theory of a norm-regulated DALMAS is presented. The algebraic model for the DALMAS is inspected and instrumentalized through an executable logic program. In particular, 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.

According to the definition of a DALMAS, an agent "enters" a set of normative positions when it is its turn to move, and this set is determined by the normative system and the actual state. Normative positions can play a role in the abstract architectures for agent systems in other ways as well. One possibility is the following. A state $s$ contains a description of all the normative positions for all agents in that state. Of course, the normative positions for an agent can be

changed when another agent moves, but an agent can only execute its rights or fulfill its duties when it is its turn to move. How the normative positions change when an agent moves is determined by the normative system for the DALMAS. (Some of the normative positions for an agent in a state can be conditional, so that they can be executed only when certain requirements are satisfied.)

## 8 Concluding remarks

Forest cleaning is a kind of activity and in the paper some aspects of the formal structures and the evaluation of activities have been discussed. The automation of forest cleaning presupposes, inter alia, principles for making "remove-decisions" and the question has been raised if such principles can be formulated as a combination of a normative system and a utility function. Some results about the representation of normative systems and the norm-regulation of multiagent-systems that may be used in the investigation of the question at issue has been outlined.

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? Of course, 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 some cases satisfying certain constraints). 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

I am very grateful to my colleagues involved in different aspects of this work: Ulla Ahonen-Jonnarth, Magnus Blom, Magnus Boman and Lars Lindahl. Financial support was given by the University of Gävle.

## References

1. Ahonen-Jonnarth U, Odelstad J. (2005). Simulation of cleaning of young forest stands. *Reports from Creativ Media Lab,* University of Gävle, Report 2005:2, 25 pp.
2. 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.

3. Blom, M. (2007). *Deontic Action-Logic Multi-Agent Systems in Prolog.* Master's Thesis, Department of Information Technology, Uppsala University.
4. Gravelle, H. & Rees, R. (1992). *Microeconomics.* Second edition, Longman, London.
5. Kanger, S. (1957) *New Foundations for Ethical Theory.* Part 1. Stockholm, 1957 (Reprinted in *Deontic Logic: Introductory and Systematic Readings,* ed. R. Hilpinen, Dordrecht, 1971, pp. 36-58.)
6. Kanger, S., Kanger, H. (1966). "Rights and Parliamentarism ", *Theoria* 32: 85-115.
7. Larsson P, Lehnbom P, Ahonen-Jonnarth U, Odelstad J. (2004). Simlation of young forest stands. Description of the program Forest stands simulation version 1.0. (In Swedish) *Reports from Creativ Media Lab,* University of Gävle, Report 2004:2, 11 pp.
8. Lindahl, L. (1977). *Position and Change. A Study in Law and Logic.* Dordrecht: Reidel.
9. Lindahl, L. & Odelstad, J. (2003). Normative Systems and Their Revision: An Algebraic Approach. *Artificial Intelligence and Law*, 11, 81-104.
10. Lindahl, L. & Odelstad, J. (2004). Normative Positions within an Algebraic Approach to Normative Systems. *Journal Of Applied Logic* 2, 63-91.
11. Lindahl, L. and Odelstad, J. (2006a). Intermediate Concepts in Normative Systems. In L. Goble and J-J. Ch. Meyer (eds.) *Deontic Logic and Artificial Normative Systems.* (DEON 2006). Berlin: Springer.
12. Lindahl, L. and Odelstad, J. (2006b). Open and Closed Intermediaries in Normative Systems. In T.M. van Engers (ed.) *Legal Knowledge and Information Systems.* (Jurix 2006). Amsterdam: IOS Press.
13. Makinson, D. (1986). On the Formal Representation of Right Relations. *Journal of Philosophical Logic* 15:403-425.
14. Odelstad, J. & Boman, M. (2004). Algebras for Agent Norm-Regulation. *Annals of Mathematics and Artificial Intelligence,* 42: 141-166, 2004.
15. Odelstad, J. & Lindahl, L. (2002). The Role of Connections as Minimal Norms in Normative Systems. *Legal Knowledge and Information Systems.* Eds. T. Bench-Capon, A. Daskalopulu and R. Winkels. Amsterdam: IOS Press.
16. Russell, S. & Norvig, P (2003). *Artificial Intelligence. A Modern Approach.* Second edition. Prentice Hall, 2003.
17. Vestlund K, Nordfjell T, Eliasson L and Karlsson A. (2005). A decision support system for selective cleaning. In: Vestlund K. 2005. *Aspects of automation of selective cleaning.* Acta Universitatits Agriculturae Sueciae 2005:74. Department of Silviculture, Swedish University of Agricultural Sciences, Umeå, Sweden. 54. p. ISBN 91-576-6973-2.

# Aligning Models of Normative Systems and Artificial Societies: Towards norm-governed behavior in virtual enterprises

**Paul Davidsson and Andreas Jacobsson**

Department of Systems and Software Engineering, School of Engineering,
Blekinge Institute of Technology, 372 25 Ronneby, Sweden
{paul.davidsson, andreas.jacobsson}@bth.se

**Abstract**. The purpose is to explore how norm-governed behavior within agent societies can be achieved in the context of Virtual Enterprises. We analyze a number of formal models from the agent research field, of which three models focus on the society aspects and three models focus on norms. A general observation is that the models reviewed are not concordant with each other and therefore require further alignment. A number of additions that may enrich the norm-focused models are suggested. It is also concluded that the introduction of different types of norms on different levels can be applied to ensure sound collaboration in agent-supported virtual enterprises. Moreover, the deployment of norm defender and promoter functionality is suggested to ensure norm compliance and punishments of norm violations.

## 1 Introduction

Artificial societies are typically characterized by agents that interact with each other in accordance with common rules or norms. Similarly to a human society, members of the artificial society must be allowed to coexist in a shared environment and to follow their respective goals in the presence of others. Here, the application of norms serves an important purpose in that they govern the rules of participation and provide important measures to achieve the desired behavior in a society.

We will here explore how norm-governed behavior within agent societies can be achieved in the context of virtual enterprises. In agent-supported virtual enterprises, the agents represent real interests and real entities, e.g., different agents have different owners, goals, interests, and preconditions for collaboration. A virtual enterprise, or more generally a Collaborative Network, may include a variety of entities (e.g., software, organizations and people) that are largely autonomous, geographically distributed, and heterogeneous in terms of their operating environment, culture, social capital, and goals. A virtual enterprise is typically described as "a temporary alliance of enterprises that join together to share skills or competencies and resources in order to better respond to business opportunities, and whose cooperation is supported by computer networks" (Camarinha-Matos et al., 2005). In this paper, a comparative study of formal models of agent societies and normative systems is undertaken in order to find out what types of norms and norm-enhancing mechanisms to include in agent-supported virtual enterprises.

In the next section, brief descriptions of the formal models reviewed are provided. This is followed by a discussion of how the various models are related to each other and of how the targeted models and the area in general can gain from model

Dagstuhl Seminar Proceedings 07122
Normative Multi-agent Systems
http://drops.dagstuhl.de/opus/volltexte/2007/908

alignment. Based on this exploration, we present some conclusions and suggestions for future work.

## 2. Formal Models

We analyze a number of formal models from the agent research field, of which three models focus on the society aspects and three models focus on norms. Each of the two groups includes two theoretical (and general) models and one specific application model.

### 2.1 Society Focus

#### *2.1.1 Theory: Artificial Societies (AS)*

Based on the work by Artikis and Pitt (2001) and Johansson (2002), Davidsson and Johansson (2006) suggest a formal characterization of agent societies that includes the following entities:

- a set of agents,
- a set of constraints on the society,
- a communication language,
- a set of roles that the agents can play,
- a set of states of affairs that hold at each time in the society,
- a set of owners (of the agents),
- a set of agent designers,
- the environment (computation and/or communication infrastructure)
- an environment owner, and
- an environment designer.

An agent is here defined as "a software entity that typically acts on the behalf of a person or an institution". Artikis and Pitt (2001) describe the set of constraints as "constraints on the agent communication, on the agent behavior that results from the social roles they occupy and on the agent behavior in general." The owner of the agent is the person or institution on whose behalf the agent acts. According to Johansson (2002) it has the power to launch the agent, provide it with preferences, as well as make run-time decisions regarding updating of preferences and when the agent should be terminated. Moreover, he defines the agent designer as the person(s) who has designed (and possibly implemented) the action selection and execution mechanisms of the agent. Davidsson and Johansson (2006) state that the environment owner is the person or organization that has the power to decide which agents may enter the society, which roles they are allowed to occupy, what communication language should be used, and the set of constraints on the society. Similarly, the environment designer is the person(s) who has designed and possibly implemented the conditions (mechanisms for controlling which agents may enter the society, what possible roles they may have, the space of constraints provided, etc.) under which the agents act in the environment.

*2.1.2 Theory: Agent-supported Virtual Enterprises (AVE)*

It has been argued that a promising approach to implement virtual enterprises is to use software agents that act as an "interface" and represent the organizations involved in a virtual enterprise. Jacobsson and Davidsson (2007) formally describe an agent-supported virtual enterprise a as a tuple:

$$ve_i = \langle A_i, R_i, AR_i, CI_i, Si, G_i, \lambda_i \rangle$$

where

- $A_i = \{a_1,..., a_n\}$ is the set of actors (typically enterprises) in $ve_i$. An actor can be described as a tuple:

  $$a_i = \langle I_i, T_i, C_i, G_i, \beta_i \rangle$$

  Where $I_i$ are the relevant information systems needed in $ve_i$, $T_i$ is the set of resources of the actor, $C_i$ is the set of core competencies of the actor, $\beta i$ is the agent acting on the behalf of the actor in $ve_i$, and $G_i$ is the set of individual goals of the actor.
- $R_i = \{r_1,..., r_m\}$ is the set of roles that the actors can play in the $ve_i$. Each actor in the virtual enterprise can play one or more roles, e.g., innovator, supplier/provider of, e.g., goods, services, expertise, etc. The choice of role depends on the virtual enterprise goal(s), the actor's core competencies, resources and individual business goals.
- $AR_i$ is a set of triples $\langle a_k, r_j, O_j^k \rangle$ where $a_k \in A$ and $r_j \in R$ i.e., the actors and their roles in the virtual enterprise, and the set of obligations, $O_j^k$, that is associated with the actor's role in the virtual enterprise.
- $CI_i$ is a set of communication infrastructures needed for operating the virtual enterprise.
- $S_i$ is a set of states of affairs that hold at each time in $ve_i$.
- $\lambda_i$ is the agent communication language used by the agents $\beta$. We will assume that $\lambda_i$ includes a set of relevant interaction protocols, a set of relevant ontologies, and possibly other things necessary to perform useful communication.
- $G_i$ is a set of goals of the virtual enterprise that is derived from the business opportunities that motivate the initiation of the virtual enterprise.

*2.1.3 Application: Plug and Play Business Communities (PPBC)*

The concept of Plug and Play Business (Jacobsson and Davidsson, 2007) supports the formation and operation of agent-supported virtual enterprises. An important concept for implementing Plug and Play Business is *Internet communities* where persons (or organizations) that share some common interests (in this case, to find partners and do business) can meet virtually. Formally, an Plug and Play Business community, $p_i$, can be described as a tuple:

$$p_i = \langle A_i, R_i, VE_i, S_i, l_i, CI_i, gk_i \rangle$$

where

- $A_i = \{a_1,..., a_n\}$ is the set of actors (typically enterprises) in the community. An actor in the Plug and Play Business community can be described as a tuple:

$$a_i = \langle I_i, T_i, C_i, G_i, h_i, b_i \rangle$$

Compared to the definition of actors in agent-supported virtual enterprises, we add $h_i$, which is the person representing the actor/enterprise, and $b_i$, which is the Plug and Play Business client software (an intelligent agent supporting the (agent) communication language, $l_i$) acting on behalf of the actor/enterprise.

- $R_i = \{r_1, ..., r_m\}$ is the set of roles that the actors can play,
- $VE_i = \{ve_1, ..., ve_l\}$ is the set of virtual enterprises currently active in the community,
- $S_i$ is a set of states of affairs that hold at any time in $p_i$,
- $l_i$ is the agent communication language used by the agents *B*. We will assume that $l_i$ includes a set of relevant interaction protocols, a set of relevant ontologies, and possibly other things necessary to perform useful communication,
- $CI_i$ is a set of communication infrastructures needed for operating the community, and
- $gk_i$ is the gate-keeper facility that regulates the entering (and leaving) of actors to (and from) the community. In order to become a member of $p_i$ there is a set of criteria that must be fulfilled, e.g., VAT number must be declared, the roles it is willing to play should be stated, and information systems must be specified. Thus, some of the aims of the gate-keeper are to ensure that this type of information is available to the Plug and Play Business community and to verify the identity of the actors. Possibly, the gate-keeper may also be equipped with capabilities of handling different levels of memberships with different sets of norms in order to cope with the varying needs of potential participants and members. The gate-keeper could also inform the potential member about what general rules that hold in the community, and require the potential member to comply with them.

Plug and Play Business supports the three critical phases of virtual enterprises: In the *definition phase*, a member of the Plug and Play Business community, typically an innovator, may at any time initiate an attempt to form a collaborative coalition in-between the members. This process may be viewed analogous to crystallization, where a catalyst (innovator) initiates a process resulting in a precise form of collaboration, i.e., the formation of a virtual enterprise. In this phase, the catalyst, $\Omega$, where $\Omega \in A$, describes the business opportunity in terms of goals, $G$ and roles, $R$, of the virtual enterprise. Since this is a highly complex task, the human representative $h_\Omega$ will be the main contributor whereas $b_\Omega$ will just support the process. Thus, in this phase the degree of norm autonomy of $b_\Omega$ is rather low, whereas in other phases it may be higher (cf. adjustable autonomy).

The *creation phase* (where crystallization takes place) consists of three subtasks and is initiated by $\Omega$:

- The function of *finding* requires that $\Omega$ has a list of the roles that must be filled in order to get an operating virtual enterprise. This list is provided by $h_\Omega$, i.e., the person representing $\Omega$ in the definition phase. Then, for each of the roles, the task for $b_\Omega$ is to find the set of candidate actors $K$ where $K \subset A$ that are able to play the role.

- In the *evaluation* task, $\Omega$ should rank the actors in $K$ according to a set of requirements $Q_r$ where $Q_r = \{q_1, q_2, ...q_k\}$ (provided by $h_\Omega$). Based on this, $\Omega$ selects the actors with the highest rank $k$ where $k \in K$ for negotiating on terms for virtual enterprise operation.
- The goal of *negotiation* is to establish an agreement between $\Omega$ and $k$ concerning $k$'s set of obligations, $O_k$. These obligations should of course be consistent with the set of goals, $G$ of the $ve$ and the set of goals of $G_k$.

When the creation phase is finished and a virtual enterprise is formed, the *operation phase* begins. Plug and Play Business supports two levels of collaboration: *administrational* and *operational*. They are defined by the type of *interaction protocols* they support. Administrational collaboration includes only protocols using the "weaker" *performatives*, such as, *ask*, *tell*, *reply*, etc. Operational collaboration supports protocols also using the performatives that actually manipulate the receiver's knowledge, such as, *insert*, where the sender requests the receiver to add the content of the message to its knowledge base, and *delete*, where the sender requests the receiver to delete the content of the message from its knowledge base.

### 2.2 Norm Focus

#### *2.2.1 Theory: Normative MAS (NMAS)*

López y López, Luck, and d'Inverno (2006) present a normative framework for agent-based systems. In their formal model, a *normative multi-agent system* consists of the following entities:

- a set of normative agents (*members*), where a normative agent consists of:
  - a set of goals,
  - a set of capabilities (actions that the agent can perform),
  - a set of motivations (preferences),
  - a set of beliefs,
  - ability to rank the goals according to preferences,
  - a set of adopted *norms*, some of which the agent has decided to comply with (*intended*) and some of which it has decided to reject (*rejected*).
- a set of general norms that govern the behavior of these agents (*generalnorms*).
- a set of norms issued to allow the creation and abolition of norms (*legislationnorms*)
- a set of norms dedicated to enforcing other norms (*enforcenorms*),
- a set of norms directed to encouraging compliance with norms through rewards (*rewardnorms*),
- the current state of the environment represented by the variable *environment*.

In addition, they identify a number of authorities:

- a set of *legislators* (agents that are entitled to create, modify, or abolish norms),
- a set of *defenders* (agents that are directly responsible for the application of punishments when norms are violated), and

- a set of *promoters* (agents whose responsibilities include rewarding compliant addressees).

The framework has been built upon the idea of autonomy of agents, i.e., it is intended to be used by agents that reason about why norms must be adopted, and why an adopted norm must be complied with.

*Norms* are formally defined to be composed of the following entities:

- a set of *normative goals*, which capture the purpose of the norm
- a set of *addressees*, which are the agents directly responsible for the satisfaction of the normative goals.
- a set of beneficiaries, which are the agents that benefit from the satisfaction of the normative goals,
- the *context*, which specifies the situations (environmental states) in which addressee agents must fulfill the norm,.
- *the exceptions*, which represent the situations in which addressees cannot be punished when they have not complied with the norm.
- *rewards* (expressed as a set of goals) to be given when normative goals become satisfied, or
- *punishments* to be applied when they are not.

*2.2.2 Theory: Normative Systems (NS)*

According to Boella and van der Torre (2004a), a normative multiagent system is composed of the following entities (we are here focusing on entities, not on how they are described, e.g., that a set of literals and rules are used to describe beliefs, desires and goals of the agents, and that there is a function, *MD*, which makes this mapping):

- a set of agents, *A*, where an agent could be either human or artificial. *A* is modeled in terms of:
  - a set of beliefs (*B*),
  - a set of desires (*D*),
  - a set of goals (*G*),
  - a set of decision variables (*X*), which represent an agent's actions,
  - a function agent description (*AD*), which maps each agent to the sets of beliefs, desires, intentions and decision variables, and
  - a priority relation (≥), which expresses each agent's characteristics and how it resolves its conflicts, i.e., rank the importance of the agent's desires and goals .
- a normative agent, **n**, which is a member of *A*,
- a set of roles, *R*, that the agents can play,
- a norm description (*V*) function that represents the norms recognized by the agents, and
- a goal distribution (*GD*) function that corresponds to the goals of the agent that it is responsible for.

Moreover, they distinguish between *regulative* norms, described as obligations, prohibitions and permissions, and *constitutive* norms, such that regulate the creation of institutional facts as well as the modification of the normative system itself. In particular, regulative norms are formalized as goals, and constitutive norms are formalized as beliefs of the normative system. Regulative norms are based on the notion of conditional obligation with an associated sanction. Obligations are defined in terms of goals of the normative agent, prohibitions are obligations concerning

negated variables, and permissions are specified as exceptions to obligations. Constitutive norms introduce new classifications of existing facts and entities, called institutional facts, or they describe the legal consequences of actions on the normative system. Roles are used to specify the powers of agents to create institutional facts or to modify the norms of the system. Thereby, constitutive norms specify both the behavior and the evolution of a system in that they introduce or remove norms from the system.

### *2.2.3 Application: Virtual Communities of Agents (VCA)*

Boella and van der Torre (2004b) investigate the use of design policies composed by prohibitions, permission and authorizations for virtual communities of agents on a computational grid. This work partly builds on Boella and van der Torre (2004A). They define a virtual community as a large, multi-institutional group of individuals who use a set of rules, i.e., a policy, to specify how to share their resources. In a virtual community, agents can play both the role of resource consumers and the role of resource providers. Resource providers retain the control of their resources and they specify in local policies the conditions for use of their resources thereby giving rise to a third role, authorization, in their model.

In virtual communities, a single set of agents ($A$), where each can play one or more roles, is defined so that each agent of the agent set can play three roles:

1. *Resource consumer*, denoted as $c(a_i)$, is an agent who manipulates a resource by means of some action. It can access resources to achieve its goals, is subject to norms regulating security, prohibitions and permissions, and also endowed with authorizations to access resources.
2. *Resource provider*, denoted as $p(a_i)$, can provide access to the resources it owns. This is referred to as the normative role, since it can issue norms, i.e., prohibitions and permissions about the access of a resource, and enforce their respect by means of sanctions, and delegate the power to authorize resource consumers.
3. *Authority*, denoted as $u(a_i)$, can declare resource consumers authorized when they are requested to do so. They know that their declarations are considered authorizations by the resource providers since they have been delegated the power to authorize resource consumers on behalf of resource providers.

*Prohibitions* and *permissions* are specified in terms of goals and desires of the bearer of the norm and of the normative role. A prohibition is defined as a goal of resource providers whereas a permission is the behavior which is not considered by a provider as a violation and thus is not sanctioned. A third concept, *authorization*, is a belief of a provider which appears as a condition in some permission it issued. Thereby, an authorization has a meaning only if it appears among the conditions of a permission. These concepts are then supplemented with two concepts, namely *violation* and *sanction*. The agent holding the normative role (i.e., the resource provider) can decide if some action is to be regarded as a violation. The possibility to punish violations by means of some sanction is among the preconditions for creating a prohibition. Sanctions provide motivation to fulfill the norms, since it is not possible to assume that all agents are cooperative and that they respect the norms. Thereby, a sanction is an action negatively affecting an agent, i.e., the agent desires the absence of the sanction.

## 3 Comparison and Model Alignment

We will now compare and try to align the different formal models described above. A more concise version of the comparison can be found in the table in Appendix A.

In this analysis, the object of study in the first set of models is the society, where norms play a small yet important part; whereas the second set of models reviewed take their starting points from normative perspectives, where the other aspects (e.g., agent ownership, agent roles, system state, etc.) play secondary roles. In the normative frameworks, different types of norms for different types of contexts are defined. By contrast, the AS model just considers one type of norms ("constraints") on only one level. Moreover, the AVE model specifies norms ("obligations") only between actors whereas the PPBC model does not include any norms at all. We argue that the use of other types of norms and on more levels than one can enrich the formal models of agent societies so that they are able to capture the types of norm-governed behavior that are necessary in many complex applications. For instance, we intend to introduce norms that regulate the interaction between the agents in the AVE as well as in the PPBC model. These can be both in terms of specific obligations or permissions between individual agents and in terms of general norms for all the agents.

Although a set of agents is defined in all the models reviewed, there are some differences in the views on what constitute an agent. Three different perspectives can be identified: (i) a norm-autonomous artificial or human entity as in the NSA and VCA models, (ii) a norm autonomous software entity as found in the NMAS model, and (iii) just a software entity. These different views obviously affect the treatment of agents and norms in artificial societies. In most current applications, agents are not norm autonomous, which may either be due to the complexity of implementing norm autonomous agents, or due to that, in some applications, it is desired that the agent owner is involved in decisions regarding what norms to follow, etc. In the PPBC model, a human representative is assisting the agent in decision situations. Moreover, it may be possible to use the theories that assume norm-autonomous entities also in current applications, but this requires that both the agent and its owner are included in the norm autonomous entity.

The agent owners are not regarded in NMAS, NS and VCA, which also may be due to that completely norm-autonomous agents are assumed in those models. Since the agent owner has the power to, apart from deciding what norms to follow, release and terminate an agent and to provide it with goals even during runtime, the inclusion of agent owners would seem to improve those formal models.

There are some suggestions on norm-enforcements amongst the reviewed frameworks. For instance, NMAS uses defender agents that are responsible for the application of punishments when norms are violated, and promoter agents that monitor norm compliance. The latter is corresponding to the external observer agent used in Kamara et al. (2005) in order to detect whether interacting agents operate in compliance with the norms or not. Moreover, they also use an admission protocol, which allows nodes to create, enter and exit the agent society. Similarly, a gate-keeper is used in PPBC to regulate the entering to and leaving from the community. A possible improvement of the PPBC would be to include also defender and promoter functionality as in the NMAS model to monitor the behavior of member actors in order to ensure norm compliance, and to impose punishments to those that violate the norms.

We can observe that an agent communication language is included only in the first set of models (AS, AVE and PPBC); however it may be implicitly assumed in the

other models. Obviously, the lack of a language would severely limit the application of norms, but even in the case where a language is used, the language may put restrictions on what norms can be expressed and communicated. However, further investigations are needed before any conclusions can be drawn.

Goals are included in all formal models except the AS model (agent preferences are discussed in the paper but not explicitly specified in the formal model). However, the goals in AVE and PPBC are not associated to agents but to actors and virtual enterprises. Since goals are closely related to norms, the introduction of norms on the agent level should be accompanied by the inclusion of goals in these models.

It can be observed that the physical environment is included in all models apart from the theoretical norm frameworks (NMAS and NS). The environment owner is only recognized in the AS model. This is interesting since the environment owner can have a large impact on what norms that hold for the society. Typically, the environment owner has control over gate-keepers, defenders, promoters, legislator, etc. Therefore, it is likely that the formal models can be enriched by including the environment owner.

## 6 Conclusions and Future Work

We have undertaken a comparative analysis of six formal models, describing artificial societies or normative systems. A general observation is that the models reviewed are not concordant with each other. For instance, completely norm-autonomous agents are assumed in the norm-focused frameworks, but in the society-focused models the notion of an agent owner is specified. In the norm-focused models, entities like agent communication language, the physical environment are often not regarded. Moreover, norm-enhancing mechanisms are only included in two models (PPBC and NMAS). Based on these findings, we have discussed how model alignment can foster both areas (modeling of agent societies and normative systems) in general and how the inclusion of missing entities may improve the various formal models in particular.

With respect to the AVE and PPBC models, we can conclude that these models can be enriched by, for instance, the introduction of norms (both on general and specific levels) that regulate the interaction between agents, and that these types of norms should be accompanied by the specification of goals. On norm enforcement, some opportunities for improvements in the PPBC model are to include defender and promoter functionality in order to ensure norm compliance and punishments of norm violations.

### 6.1 Future Work

Based on our analysis, there are a number of issues that need further study, e.g.:

- Investigate what types of norms (obligations, permissions, prohibitions, etc.) and on what levels (general or specific) to include in the formal models of both AVE and PPBC.
- Explore the possibility to include the owner in norm-autonomous entities in the NMAS, NS and VCA models, so that these models can be applied also to societies populated by agents that are not norm-autonomous.
- A possible improvement of the PPBC would be to include also defender and promoter functionality as in the NMAS model.

## References


Artikis, A, and Pitt, J (2001) A Formal Model of Open Agent Societies. In: Proceedings of the 5th International Conference on Autonomous Agents.

Boella, G, and van der Torre, L (2004a) Regulative and Constitutive Norms in Normative Multiagent Systems. In: Proceedings of 9th International Conference on the Principles of Knowledge Representation and Reasoning (American Association for Artificial Intelligence).

Boella, G, and van der Torre, L (2004b) Virtual Permission and authorization in policies for virtual communities of agents. In: Proceedings. of Agents and P2P Computing Workshop at AAMAS'04.

Camarinha-Matos, LM, and Afsarmanesh, H (2005) Collaborative Networks: A New Scientific Discipline. In: Journal of Intelligent Manufacturing, 16: 439-452.

Davidsson, P (2001) Categories of Artificial Societies. Engineering Societies in the Agents World II, Lecture Notes in Computer Science, Vol. 2203, Springer Verlag, Berlin Germany.

Davidsson, P, and Johansson, S (2006) On the Potential of Norm-Governed Behavior in Different Categories of Artificial Societies. In: Computational and Mathematical Organization Theory, 12:169-180.

Jacobsson, A and Davidsson, P (2007) A Formal Analysis of Virtual Enterprise Creation and Operation. To appear in eds. Król, D, and Nguyen, NT: Intelligence Integration in Distributed Knowledge Management by Idea Group Publishing.

Johansson, SJ (2002) On Coordination in Multi-Agent Systems. Ph.D. Dissertation Series No. 05/02., Department of Software Engineering and Computer Science, Blekinge Institute of Technology, Sweden.

Kamara, L, Pitt, J, and Sergot, M (2005) Towards Norm-Governed Self-Organizing Networks. In: 1st International Symposium on Normative Multiagent Systems.

López y López, F, Luck, M, and d'Inverno, M (2006) A Normative Framework for Agent-Based Systems. In: Computational and Mathematical Organization Theory, 12:227-250.

## Appendix A

| | **Theory: AS** | **Theory: AVE** | **Application: PPBC** | **Theory: NMAS** | **Theory: NS** | **Application: VCA** |
|---|---|---|---|---|---|---|
| **Agents** | A set of agents | A set of agents (*B*) | A set of agents (*B*) | A set of normative agents (*members*) | A set of agents (A) | A set of *n* agents (*A*) |
| **Norms** | A set of constraints | A set of obligations for each role that the actors can play ($O^{role}$) thus this is not on the agent level | ----- | Four sets of norms: *generalnorms*, *legislationnorms*, *enforcementnorms*, and *rewardnorms* (which in turn may be obligations, prohibitions, social commitments or social codes) | Four sets of norms: obligations, permissions, prohibitions, and institutional facts | Three sets of norms: prohibitions, permissions and authorizations |
| **Communication language** | A communication language | $\lambda_i$ is the agent communication language used by the agents *B* | $l_i$ is the agent communication language used by the agents *B* | ----- | ----- | ----- |
| **Agent roles** | A set of roles that the agents can play | The role of participant is implicitly assumed | The roles of initiator and potential participant are implicitly assumed | With respect to a norm, an agent can play either the *addressee* or the *beneficiary* | A set of roles, *R*, that the agents can play | An agent can play three roles *resource consumer*, *resource provider* or *authority* |
| **System state** | A set of states of affairs that hold at each time at the society | $S_i$ is a set of states of affairs that hold at each time in the virtual enterprise | $S_i$ is a set of states of affairs that hold at each time in the Plug and Play Business community | The current state of the environment is represented by the variable *environment* | *Parameters* (*P*) describe both the state of the world and *institutional facts* | ----- |
| **Agent owners** | A set of owners (of the agents) | A set of actors (*A*) | A set of actors (A), which have human representatives (*H*) | ----- | ----- | ----- |
| **Agent owner roles** | ----- | A set of roles (*R*) | A set of roles (*R*) | ----- | ----- | ----- |
| **The Physical Environment / Infrastructure** | The environment (computation and/or communication infrastructure) | A set of communication infrastructures needed for formation and collaboration (*CI*) | A set of communication infrastructures needed for formation and collaboration (*CI*) | ----- | ----- | A grid infrastructure is used for the virtual communities |
| **Norm-enhancing mechanisms** | ----- | ----- | Gate-keeper agent (regulates the entering and leaving to and from the community) | *Defenders* (agents that are responsible for the application of punishments when norms are violated) *Promoters* (agents that monitor compliance with norms) *Legislators* (agents that define norms) | ----- | ----- |
| **Environment owners** | An environment owner | ----- | ----- | ----- | ----- | ----- |
| **Goals** | ----- | Each VE has a set of goals ($G_{VE}$) and each actor has a set of goals ($G_{actor}$) | Each VE has a set of goals ($G_{VE}$) and each actor has a set of goals ($G_{actor}$) | Each agent has a set of goals (*goals*) | Each agent has a set of beliefs ($B_a$), desires ($D_a$), and goals ($G_a$) | Each agent has a set of beliefs ($B_a$), desires ($D_a$), and goals ($G_a$) |

| | **Theory: AS** | **Theory: AVE** | **Application: PPBC** | **Theory: NMAS** | **Theory: NS** | **Application: VCA** |
|---|---|---|---|---|---|---|
| **Definition of agent** | A software entity that typically acts on the behalf of a person or an institution | ----- | ----- | Normative agent:<br>- a set of goals<br>- a set of capabilities<br>- a set of motivations (preferences)<br>- a set of beliefs<br>- ability to rank the goals according to preferences<br>- a set of norms<br>- a set of intended norms<br>- a set of rejected norms | Human or artificial. An agent is defined as:<br>-Beliefs (B)<br>-Desires (D)<br>-Goals (G)<br>-Decision variables (X)<br>-Agent description variables (AD)<br>-A priority relation (≥) | Agent design is inspired by the BOID architecture |
| **Definition of norm** | Constraints on the agent communication language, on the agent behavior that results from the social roles they occupy, and on the agent behavior in general | ----- | ----- | Components are:<br>-Normative goals<br>-Addressee agents<br>-Beneficiary agent<br>-Context<br>-Exception<br>-Have not (complied with norms)<br>-Immunity<br>-Rewards<br>-Punishments | Norms are either *regulative* (defined as obligations, prohibitions, and permissions) or *constitutive* (defined as institutional facts) | Prohibitions and permissions are defined in terms of goals and desires of the bearer of the norm and of the normative role, together with the concepts of violation and sanction. |

# BIO Logical Agents: Norms, Beliefs, Intentions in Defeasible Logic

Guido Governatori[1] and Antonino Rotolo[2]

[1] School of Information Technology and Electrical Engineering, The University of Queensland,
Brisbane, Queensland, QLD 4072, Australia
guido@itee.uq.edu.au
[2] CIRSFID and Law Faculty, University of Bologna
Via Galliera 3, 40121, Bologna, Italy
rotolo@cirsfid.unibo.it

**Abstract.** In this paper we follow the BOID (Belief, Obligation, Intention, Desire) architecture to describe agents and agent types in Defeasible Logic. We argue, in particular, that the introduction of obligations can provide a new reading of the concepts of intention and intentionality. Then we examine the notion of social agent (i.e., an agent where obligations prevail over intentions) and discuss some computational and philosophical issues related to it. We show that the notion of social agent either requires more complex computations or has some philosophical drawbacks.

**Keywords.** Social Agents, Defeasible Logic, Complexity of Agents

## 1 Introduction

Reasoning about mental attitudes is a traditional issue in philosophy and has widely investigated in the field of AI. Some classical agent systems based on mental attitudes such as beliefs, desires and intentions are, for example, those presented in [1,2,3].

More recent works on cognitive agents tried combine two apparently independent perspectives [4,5,6,7,8,9]: (a) a classical cognitive account of agents that specifies their mental attitudes; (b) modelling agents' behaviour by means of normative concepts. For the first approach, the background is basically the belief-desire-intention (BDI) architecture, where mental attitudes are taken as primitives to give rise to a set of Intentional Agent Systems [3,1]. This view is interesting especially when the behaviour of agents is the outcome of a rational balance among their (possibly conflicting) mental states. The normative aspect is rather based on the assumption that normative concepts play a role to characterise the idea of social co-ordination of autonomous agents [10]. The nice result of this combination of perspectives is that of leading to an account of agents' deliberation and behaviour in terms of the interplay between mental attitudes and normative (external) factors such as obligations.

Dagstuhl Seminar Proceedings 07122
Normative Multi-agent Systems
http://drops.dagstuhl.de/opus/volltexte/2007/912

A crucial aspect in this recent trend is that reasoning about agents can be embedded in frameworks based on non-monotonic logics, as one the most interesting problems concerns the cases where the agent's mental attitudes are in conflict or when they are incompatible with obligations and other deontic provisions. In this specific perspective, the relation between mental attitudes and non-monotonicity should not sound surprising: works such as Thomason's [11] and on BOID [4] confirm this trend. Of particular interest is the BOID architecture, which in fact provides a number of strategies for solving conflicts among mental attitudes and obligations. BOID specifies logical criteria (i) to retract agent's attitudes with the changing environment, and so (ii) to settle conflicts by stating different general policies corresponding to the agent type considered. Agent types correspond to the different ways through which conflicts are detected and solved: a realistic agent thus corresponds to a conflict-resolution type in which beliefs override all other factors, while other agent types, such as simple-minded, selfish or social ones adopt different orders of overruling.

Following [5,6,7], in this paper we take advantage of this research line and discuss how the combination of mental attitudes and obligations can be framed in Defeasible Logic (DL). As is well-known, DL is based on a logic programming-like language and it is a simple, efficient but flexible non-monotonic formalism able to deal with many different intuitions of non-monotonic reasoning and recently applied in many fields. In addition, several efficient implementations have been developed [12,13]. Here we discuss and extend some aspects of a non-monotonic logic of agency, based on the framework of [14], developed in [6,7]. Indeed, DL is one of the most expressive languages that allows for the definition of large sets of patterns called agent types.

Our system, which considers here three components –beliefs, intentions, and obligatios (BIO agents)– has some substantial peculiarities that make it different from other frameworks such as BOID's. In particular,

- the system develops a constructive account of those modalities that correspond to mental states and obligations; rules are thus meant to devise suitable logical conditions for introducing modalities; if so, rules may also contain modalised literals;
- possible conversions of a modality into another can be accepted, as when the applicability of rule leading to derive, fro example, OBL$p$ ($p$ is obligatory) may permit, under appropriate conditions, to obtain INT$p$ ($p$ is intended).

Both aspects are necessary to account for some relatively simple, but important reasoning patterns. In particular, we maintain that conversions are required to capture some aspects of agents' rationality.

A second result of this paper consists in showing that the proposed system is computationally feasible. Indeed, we will prove that it is possible to compute the complete set of consequences of a given theory in linear time, thus preserving the nice computational features of standard DL.

However, despite the computational feasibiliy of the logic, we will argue that the notion of agent type is problematic. Here, the focus will be in particular on philosophical and computational aspects of the notion of social agent, by which

we mean a norm-complying agent, but we will argue that similar considerations also apply to other agent types.

The layout of the paper is as follows. Section 2 provides the theoretical background of our system. In particular, since the concept of social agent focuses on the interplay between obligations and intentions, we will discuss which kind of intentions have to be considered in this regard. Section 3 will present our logical framework, based on DL, which will embed our intuitions and permit to deal with BIO agents. Section 4 presents a first discussion of the notion of agent type; in particular, we will argue that conversions, too, are relevant in identifying specific cognitive profiles for the agents; the section ends with an open problem concerning the feasibility of agent types based on the strategies for solving conflicts. Section 5 deals with the computational complexity of social agency. A concluding section on related work completes the paper.

## 2 Norms, Beliefs and Intentions

### 2.1 Policy-based Motivations

This section provides some theoretical background for the rest of the paper. Our focus is on the so-called policy-based attitudes. The term was coined by Bratman [15] with specific reference to the idea of intention. The intuition behind policy-based intention is based on Bratman's view regarding *future directed intention* and *general intention*. Bratman terms general intentions as *general/personal policies*. Along with general policies go policy-based intentions. For example, I have a general policy to patch up and reboot the Unix server in the department once every month. This morning, on the basis of this policy, I form the intention to reboot the machine at 7.00 pm in the evening. My intention this morning to reboot the machine this evening is a *policy-based* intention. This specific intention will play a major part in my planning process for the day as it will pose problems about means and constrain my other options. Based on this distinction Bratman makes the following classification of intentions: *deliberative, non-deliberative* and *policy-based*.

The difference between the three is the following: When an agent $i$ has an intention of the form $\mathrm{INT}_i^{t_1}\varphi, t_2$ (read as *agent $i$ intends at $t_1$ to $\varphi$ at $t_2$*) as a process of *present* deliberation, then it is called *deliberative intention*. On the other hand, if the agent comes to have such an intention not on the basis of present deliberation, but at some earlier time $t_0$ and has retained it from $t_0$ to $t_1$ without reconsidering it, then this intention it is called *non-deliberative*. The third case arises when intentions are general and concern potentially recurring circumstances in an agent's life. Such general intentions constitute *policy-based intentions*. A policy-based intention is not a non-deliberative intention because it is not simply a case of retaining an intention previously formed. Neither is it a deliberative intention since it is not based on a full-blown deliberation where an attempt is made to weigh pros and cons for and against conflicting options. It also differs from an intention in favour of necessary means, i.e., intention in favour of a specific end, in the sense that the defeasibility of general policies

makes it possible to *block* the application of the policy to the particular case without *abandoning* the policy. Otherwise one could abandon the intention in favour of the end. The difference here is that in each case the policy concerns not just a single future situation, but a kind of circumstance that is expected to recur in the agent loop and in each case the agent might well have a general intention to act in the particular circumstances. Whether the agent is able to perform that action or not depends on the circumstances.

As argued in detail elsewhere [16], it may happen that a policy-based intention needs to be *re-considered* if not *blocked* for the application to particular cases. But this does not mean that the agent should know all such conditions in a scenario, but only those she considers necessary for the intended outcome and that she is not confident of their being satisfied. To intend the necessary consequence the agent has to make sure that all the evidence to the contrary has been defeated, which is basically a defeasible conclusion.

The starting point of this paper is to extend the policy-based approach to other attitudes and motivational factors such as beliefs and obligations. In this way, all motivational factors are represented within a rule-based system: intentions and beliefs are viewed as constituting the internal constraints (based on policies) of an agent while obligations are her external constraints (based on rules). As constraints they are defeasible. Notice, in particular, that such an extension to obligations can capture the well-known defeasible character of deontic reasoning. In this last case, a policy-based obligation –conceived of as an external motivational attitude– turns out to be simply a conditional obligation, namely, a rule that allows for the inference of an obligation whenever the antecedent of this rule holds [17,18].

### 2.2 Expected Side Effects and Agents' Rationality

According to Bratman, rational agents can be basically modelled as follows:

- agents are goal-directed without being necessarily aware of their activity;
- intentions are used to choose partial plans for the realisation of a goal;
- not all consequences are intended but only some initial intentions and the goal as a result of the plan; if some side-effects occur, they are never intended.

It is worthy discussing here the notion of side-effect. This well-known problem has to deal with several variants of logical omniscience: the problem arises when the agent is required to know all the truths defined by her logic, or when the logic that depicts the agent automatically includes all the logical truths of classical logic, or, finally, if the agent knows all the logical consequences of the known propositions [19]. Indeed, the problem is usually referred to as the *expected side-effects* problem [15], a problem which depends on the interactions between the reasoning mechanism for the propositional inferences and the mechanism ruling the introduction and the behaviour of the modal operators. A simple and rather unsatisfactory solution would be to consider two completely unrelated consequence relations, one for the propositional part and the second one for the

modal operators. The consequence relation for a modal operator is meant to give the condition under which one can prove a modal formula. For example the pair $\Gamma \mid\sim_X \alpha$, where $X$ is a modal operator, means that if we can prove all the formulas in $\Gamma$ then we can deduce $X\alpha$. In what follows we will develop a system for mental states and motivational attitudes based on this idea. However, we will allow the consequence relation for intentions and obligations to interact with the propositional module and we will also consider possible interactions between the modal operators. To this end we have to show that the expected side-effects phenomenon is not a drawback for policy-based agents: such a kind of agents must accept the expected-side effects unless they have some reasons to reject the consequences corresponding to them.

In effect, though our proposed theory does not entertain many of the properties leading to logical omniscience, some aspects of the side-effects problem are accepted. Consider

$$\mathrm{INT}\,Smoke,\ Smoke \rightarrow Cancer \mid\sim \mathrm{INT}\,Cancer \tag{1}$$

$$\mathrm{INT}\,GoToRome,\ GoToRome \rightarrow GoToItaly \mid\sim \mathrm{INT}\,GoToItaly \tag{2}$$

Actually, whereas the first case is clearly unacceptable, the second should be accepted by a rational agent. In this perspective the side-effects problem is similar to the substitution of indiscernible in opaque contexts. An agent may have the intention to visit Rome and not to visit Italy. But if the agent knows that Rome is the capital of Italy then it would be irrational of the agent not to have the intention to go to Italy given the intention to visit Rome.

Accordingly, some cases of the side-effects problem are not necessarily a weakness of a theory. This holds in particular if we assume that our agents are *aware* of their activities. In our view, modelling rational agents corresponds to the following assumptions:

- agents are aware of their activities, of their policies;
- some cases of the side-effects problem can be accepted;
- if a case has to be rejected this means that its unpleasant consequences should not be intended;
- when unpleasant consequences are not intended, this only means that they are blocked by conflicting attitudes or facts.

The theory an agent is equipped with can be understood as the specification of the behaviour of the agent. If the agent is *aware* that $B$ is an unavoidable/indisputable consequence of $A$ and the agent intends $A$, then $B$ is a consequence of the agent's intentions and the agent must accept it as part of her intentions. Suppose we have that "raising one's hand at an auction counts as making a bid". Thus if the agent (aware of this policy) intends to raise her hand, then she intends to bid in the auction, and her action will be understood as making a bid. In other words, in our system we will try to balance and moderate some unpleasant aspects of the side-effects problem with the equally important need for modelling rational agents. Of course, according to our view, we may have that something is intended even if it is causally distant with respect to the

original derived intentions. But this is not necessarily a drawback if we conceive agents as rational and, as such, being aware of the policies which are related with the environment and with their interests: even a causally distant behaviour can be rationally intended unless it is removed in the meantime from deliberation. But this case is indeed considered within our analysis because we may have concrete contexts in which some policy-based intentions, as soon as they are applicable, turn out to be overridden by other policies: we may have reasons to argue that, if an agent intends $A$ and believes that $B$ is a consequence of $A$, this is not a reason for necessarily intending $B$; in fact, the derivation of $B$ as an intention may be blocked, in our view, by competing attitudes or made non-applicable by concrete facts.

According to the previous discussion it should be clear that, though inspired by Bratman's [15] analysis, the notion of intention we study in this paper is slightly different, as it focuses on the idea of *intentionality*. In Bratman's view intentions are used to choose partial plans for the realisation of a goal; in this way they have a close relation to means-ends. In our view intentions should be related not only to means-ends but also to the their consequences.

This concept of intention is particularly relevant in conjunction with deontic and normative notions, for example if we want to say that an agent is legally responsible for $A$ if the agent did $A$ with the intention to do $A$. In such cases the agent has to include in the set of her intentions not only her intentions in Bratman's sense but also some of their consequences. Our intuition is compatible with von Wright's [20] classical theory of normative actions. Von Wright's problem is to identify what should be the content of norms. He argues that norms should deal with actions. Roughly, actions can be described in terms of state transitions and as the sets of all changes of world that follow from them. It is not our purpose discussing here von Wright's theory of action. It should be noted, however, that he considers the related problem of intentions. On the one hand, von Wright is clear when he says that any action may have an arbitrary number of consequences and not all of them are intended. On the other hand, he provides a very broad concept of action, according to which all actions in norms, strictly speaking, are intentional. If so, what are the boundaries of intentions to be considered when they interplay with obligations?

Let us see how to recast Bratman's Strategic Bomber scenario [15] in this perspective. The basic scenario runs as follows: Strategic Bomber intends to bomb a munition plant of the enemy being aware that the resulting explosion will kill innocent children in a nearby school. Bratman argues that Strategic Bomber does not have the intention to kill the children. Let us expand the scenario by supposing that despite the bombing, Strategic Bomber loses the war, and that there is a process for war crimes against him. Civil casualties are a sad but almost unavoidable consequence of war, but usually the killing of civilians does not constitute a war crime if there was no intention to kill. According to Bratman, Strategic Bomber did not commit a war crime since he did not have such an intention. However, let us assume that Strategic Bomber did not do anything to prevent or minimise civil casualties (let us say by a movement of

troops that might have resulted in an evacuation of the area surrounding the munition plant). In this extended scenario the killing of children is brought about by a (successful) intentional act of Strategic Bomber. Accordingly, he must be held responsible for the killing of innocent civilians.

Given this interpretation of intentions, we will see in the rest of this paper that some standard accounts of agent types, and of social agents in particular, are not satisfactory.

## 3 BIO Agents in Defeasible Logic

### 3.1 Basics of Defeasible Logic

Defeasible Logic (DL) was originally proposed by Nute [21,22] with a particular concern about computational efficiency and developed over the years notably by [23,24,14]. DL is suitable for implementations [25], is flexible [14] (it has a constructively defined and easy to use proof theory), and it is modular [24] (it can be easily extended to cover different logical components: besides the current contribution, see, e.g., [6,7]). In addition, DL is efficient: it is possible to compute the complete set of consequences of a given theory in linear time [26]. As we will see, this result also applies to the logical framework presented in this paper.

Knowledge in DL can be represented in two ways: facts and rules.

*Facts* are indisputable statements and are represented by predicates. We only use a propositional language. Facts containing free variables are interpreted as the set of their variable-free instances. For example, "the price of the spam filter is \$50" is represented by $Price(SpamFilter, 50)$.

A *rule*, on the other hand, describes the relationship between a set of literals (premises) and a literal (conclusion), and we can specify how strong the relationship is. As usual, rules allow us to derive new conclusions given a set of premises. As far as the strength of rules is concerned we distinguish between *strict rules*, *defeasible rules* and *defeaters*.

Strict rules, defeasible rules and defeaters are represented, respectively, by expressions of the form $A_1, \ldots, A_n \rightarrow B$, $A_1, \ldots, A_n \Rightarrow B$ and $A_1, \ldots, A_n \rightsquigarrow B$, where $A_1, \ldots, A_n$ is a possibly empty set of prerequisites and $B$ is the conclusion of the rule. We only consider rules that are essentially propositional. Rules containing free variables are interpreted as the set of their ground instances.

*Strict rules* are rules in the classical sense: whenever the premises are indisputable then so is the conclusion. Thus they can be used for definitional clauses. An example of a strict rule is "A 'Premium Customer' is a customer who has spent \$10,000 on goods":

$$TotalExpense(x,\ 10000) \rightarrow PremiumCustomer(x).$$

*Defeasible rules* are rules that can be defeated by contrary evidence. An example of such a rule is "Premium Customer are entitled to a 5% discount":

$$PremiumCustomer(x) \Rightarrow Discount(x).$$

The idea is that if we know that someone is a Premium Customer, then we may conclude that she is entitled to a discount *unless there is other evidence suggesting that she may not be* (for example if she buys a good in promotion).

*Defeaters* are a special kind of rules. They are used to prevent conclusions not to support them. For example:

$$SpecialOrder(x), PremiumCustomer(x) \rightsquigarrow \neg Surcharge(x).$$

This rule states that premium customers placing special orders might be exempt from the special order surcharge. This rule can prevent the derivation of a "surcharge" conclusion. On the other hand it cannot be used to support a "not surcharge" conclusion.

DL is a "skeptical" non-monotonic logic, meaning that it does not support contradictory conclusions. Instead DL seeks to resolve conflicts. In cases where there is some support for concluding $A$ but also support for concluding $\neg A$, DL 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.

As we have alluded to above, no conclusion can be drawn from conflicting rules in DL unless these rules are prioritised. The *superiority relation* among rules is used to define priorities among rules, that is, where one rule may override the conclusion of another rule. For example, given the defeasible rules

$$r : PremiumCustomer(x) \Rightarrow Discount(x)$$
$$r' : SpecialOrder(x) \Rightarrow \neg Discount(x)$$

which contradict one another, no conclusive decision can be made about whether a Premium Customer who has placed a special order is entitled to the 5% discount. But if we introduce a superiority relation $>$ with $r' > r$, then we can indeed conclude that special orders are not subject to discount.

Informally, conclusions can be drawn in DL according to the following intution. Let $D$ be a theory in DL (i.e., a collection of facts, rules and a superiority relation over the set of rules). A *conclusion* of $D$ is a tagged literal and can have one of the following four forms:

$+\Delta q$ meaning that $q$ is definitely provable in $D$ (i.e., using only facts and strict rules).
$-\Delta q$ meaning that we have proved that $q$ is not definitely provable in $D$.
$+\partial q$ meaning that $q$ is defeasibly provable in $D$.
$-\partial q$ meaning that we have proved that $q$ is not defeasibly provable in $D$.

Strict derivations are obtained by forward chaining of strict rules, while a defeasible conclusion $p$ can be derived if there is a rule whose conclusion is $p$, 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 $\neg p$ has prerequisites that fail to be derived. In other words, a conclusion $p$ is derivable when:

– $p$ is a fact; or

– there is an applicable strict or defeasible rule for $p$, and either
  - all the rules for $\neg p$ are discarded (i.e., not applicable) or
  - every applicable rule for $\neg p$ is weaker than an applicable strict[3] or defeasible rule for $p$.

In the next sections we will see how the basic machinery of DL can be extended to deal with the multi-modal logic required to model BIO agents.

### 3.2 Modal Defeasible Logic

Our purpose is to account for policy-based motivations of BIO agents, which requires to capture at least some basic facets of the modal notions of belief, intention, and obligation.

Usually modal logics are extensions of classical propositional logic with some intensional operators. Thus any modal logic should account for two components: (1) the underlying logical structure of the propositional base and (2) the logic behaviour of the modal operators. Alas, as is well-known, classical propositional logic is not well suited to deal with real life scenarios. The main reason is that the descriptions of real-life cases are, very often, partial and somewhat unreliable. Our discussion in Section 2 is in line with this intution as far as agents' motivational attitudes are concerned. Accordingly, in such circumstances classical propositional logic might produce counterintuitive results insofar as it requires complete, consistent and reliable information. Hence any modal logic based on classical propositional logic is doomed to suffer from the same problems.

On the other hand the logic should specify how modalities can be introduced and manipulated. Some common rules for modalities are, e.g.,

$$\frac{\vdash \varphi}{\vdash \Box\varphi} \text{ Necessitation} \qquad \frac{\vdash \varphi \supset \psi}{\vdash \Box\varphi \supset \Box\psi} \text{ RM}$$

Both dictates conditions to introduce modalities based purely on the derivability and structure of the antecedent. These inference rules are related to the mentioned problem of logical omniscience: if $\Box$ corresponds either to INT, BEL, or OBL, they put unrealistic assumptions on the cognitive capabilities of an agent. In effect, although some aspects of the expected side-effects problem should be accepted in modelling rational agents, rules such as Necessitation and RM are clearly too demanding: both in general permit to derive that an agent believes or intends something, or that something is obligatory for her, assuming that she knows all the truths defined by her logic, or that the logic that depicts her behaviour automatically includes all the logical truths of classical logic, or that she knows all the logical consequences of known propositions.

The point is thus avoid these difficulties by only admitting the side effects for which no contrary reason can be advanced. Our strategy is twofold. First, we take a constructive interpretation of $\Box$: we have that if an agent can build

[3] Notice that a strict rule can be defeated only when its antecedent is defeasibly provable.

a derivation of $\varphi$ then she can build a derivation of $\Box\varphi$. We want to maintain this intuition, but also to replace derivability in classical logic with a practical and feasible notion like derivability in DL. Thus the intuition behind this work is that we are allowed to derive $\Box p$ if we can prove $p$ with the mode $\Box$ in DL.

To extend DL with modal operators we have two options: 1) to use the same inferential mechanism as basic DL and to represent explicitly the modal operators in the conclusion of rules [27]; 2) introduce new types of rules for the modal operators to differentiate between modal and factual rules.

For example the "deontic" statement "The Purchaser shall follow the Supplier price lists" can be represented as

$$AdvertisedPrice(X) \Rightarrow \mathrm{OBL}_{purchaser} Pay(X)$$

if we follow the first option and

$$AdvertisedPrice(X) \Rightarrow_{\mathrm{OBL}_{purchaser}} Pay(X)$$

according to the second option, where $\Rightarrow_{\mathrm{OBL}_{purchaser}}$ denotes a new type of defeasible rule relative to the modal operator $\mathrm{OBL}_{purchaser}$.

The differences between the two approaches, besides the fact that in the first approach there is only one type of rules while the second accounts for factual and modal rules, is that the first approach has to introduce an additional machinery for introducing and reasoning with modal operators. Hence, explicitly representing the modal operators in the conclusion of rules does not follow the basic intution we have suggested above. In fact, in this case we would have to provide a definition of $p$-incompatible literals (i.e., a set of literals that cannot be hold when $p$ holds.) for every literal $p$. For example we can have a modal logic where $\Box p$ and $\neg p$ cannot be both true at the same time. Moreover the first approach is less flexible than the second: in particular in some cases it must account for rules to derive $\Diamond p$ from $\Box p$; similarly conversions –which permit to use a rule for a certain modality as it were for another modality (see *infra*)– require additional operational rules in a theory, thus the second approach seems to offer a more conceptual tool than the first one. It seems that the second approach can use different proof conditions based on the modal rules to offer a more fine grained control over the modal operators and it allows for interaction between modal operators.

If we label the arrows of the rules (i.e., agent's policies) of our rule-based system by the different modalities we want to deal with, then this solution leads to distinguishing different modes through which the literals can be derived using rules. How such types of derivation are related to the introduction of the corresponding modalised literals can be expressed as follows: if $X \in \{\mathrm{BEL}, \mathrm{INT}, \mathrm{OBL}\}$, then

$$\frac{\Gamma \quad \Gamma \Rightarrow_X \psi}{\Gamma \mid\sim X\psi} \ \mathrm{MI}$$

As we will see, we do make an exception when rules for belief are concerned since we will state that $X \in \{\mathrm{INT}, \mathrm{OBL}\}$. The reason for this is that we assume that

beliefs are conceived of as the knowledge the agent has of the enviroment, and so they are used by the agent to make inferences about how the world is: in this perspective, belief conclusions correspond to factual knowledge and do not need to be modalised. But besides this exception, which can be removed if required, schema MI captures the basic logical behaviour of our modal rules.

So far, so good. However, if nothing is done besides labelling the rules of DL, what we have in our hands is nothing but a simple treatment of modalities: what we obtain is that the conditions for introducing modalities (and in particular intentions and obligations) collapse into those for deriving literals in standard DL. Hence, the next step is to allow the consequence relations to interact with the propositional module and with each other. Indeed, we could in theory define sets of many interaction patterns, but what we need for the purposes of our paper are only two interaction strategies: one that permits to use rules for a modality $X$ as they were for another modality $Y$ (*rule conversions*), and one that considers conflicts between rules (*conflict-detection* and *conflict-resolution*).

*Rule Conversions* The notion of *rule conversion* allows us to model peculiar interactions between different modal operators. In general, notice that in many formalisms it is possible to convert from one type of conclusion into a different one. Take for example the right weakening rule of non-monotonic consequence relations (see, for example [28])

$$\frac{B \vdash C \quad A \mid\sim B}{A \mid\sim C}$$

which allows the combination of non-monotonic and classical consequences.

Suppose that a rule of a specific type is given and also suppose that all the literals in the antecedent of a rule are provable in one and the same modality. If so, is it possible to argue that the conclusion of the rule inherits the modality of the antecedent? To give an example, suppose we have that $\psi \Rightarrow_{\mathrm{BEL}} \phi$ and that we derive $\psi$ using a rule labelled by INT. Can we conclude $\mathrm{INT}\phi$? If the answer is positive, on the basis of MI this can be represented as follows:

$$\frac{\Gamma \mid\sim \mathrm{INT}\psi \quad \psi \Rightarrow_{\mathrm{BEL}} \phi}{\Gamma, \mathrm{INT}\psi \mid\sim \mathrm{INT}\phi} \text{ Conversion}$$

In many cases this is a reasonable conclusion to obtain. Indeed, this is the inference pattern we discussed in Section 2: if an agent believes to visit Italy if she visits Rome, and she has the intention to visit Rome, then it seems rational that she has the intention to visit Italy. Thus, conversions are ways through which some rational side effects can be derived. An additional example can help us illustrate the notion of conversion. Consider the following formalisation of the Yale Shooting Problem.[4]

$$load\_live\_ammo, shoot \Rightarrow_{\mathrm{BEL}} kill$$

[4] Here we will ignore all temporal aspects and we will assume that the sequence of actions is done in the correct order.

This rule encodes the knowledge of an agent that knows that loading the gun with live ammunitions, and then shooting will kill her friend. This example clearly shows that the qualification of the conclusions depends on the modalities relative to the individual acts "load" and "shoot". In particular, if we obtain that the agent intends to load and to shoot the gun (INT($load$), INT($shoot$)), then, since she knows that the consequence of these actions is the death of her friend, she intends to kill him. However, if shooting was not intended, then we have prima facie to say that killing, too, was not intentional.

To define the admitted conversions we introduce a binary relation Convert over the modalities of the language. When we write Convert(BEL, INT) this means that a belief rule $r$ can be used to derive an intention (of course, provided that all its antecedents are derived as intentions): $r$ can thus be converted into a rule for intention. Notice that we do not impose any specific constraint on Convert. In particular, we do *not* require Convert to be irreflexive. In fact, rule conversions can be viewed as corresponding, in a multi-modal setting, to the following inference schema:

$$\frac{X\psi \quad Y(\psi \to \phi)}{X\phi} \tag{3}$$

If we have Convert$(X, Y)$ and $X = Y$, we do not obtain something necessarily odd. As is well-known, in deontic logic, for example, this inference pattern corresponds to the so-called deontic detachment:

$$\frac{\text{OBL}\psi \quad \text{OBL}(\psi \to \phi)}{\text{OBL}\phi} \tag{4}$$

Although (4) is far from being uncontroversial, it seems that the same philosophical reasons that lead to accept it may support, for example, the adoption of its counterpart for intentions. Thus, even though we do not want in general to accept (3) when $X = Y$, we believe that this case cannot be excluded, and so, a fortiori, that Convert$(X, X)$ be always rejected.

*Conflicts* As was mentioned in the previous sections, conflict-detection and conflict-resolution play an important role in the current context. It is in fact crucial to establish criteria for detecting and solving conflicts between the different components which characterise the cognitive profiles of agent's deliberation. In a multi-modal setting, we can establish which modalities can be incompatible with each other, and, also, we can impose various forms of consistency, such as the following:

$$X\phi \to \neg Y\neg\phi \tag{5}$$

$$(X\phi \wedge Y\neg\phi) \to \neg Z\neg\phi \tag{6}$$

Criteria for conflict-detection and -resolution in DL can capture the rationale of schemata such as (5) and (6). However, their precise definition makes it necessary to take care of the peculiar approach adopted. In particular, various forms

of consistency between agents' motivations require to define incompatibility relations between the modalities by referring to rule types as well as to specific methods to solve conflicts between the rules. Many complex conflict patterns can be identified [5,7,6]. For the purpose of this paper, we introduce a binary and asymmetric relation Conflict over the set of modalities that defines which types of rules are in conflict and which are the stronger ones (the formal definition of Conflict is given in Section 3.3). Suppose, for example, that we have

$$r : a \Rightarrow_{\text{BEL}} q$$
$$s : b \Rightarrow_{\text{OBL}} \neg q$$
$$t : c \Rightarrow_{\text{INT}} q$$

If we only have Conflict(BEL, OBL), this means that rule $r$ is in conflict with rule $s$ and that $r$ is stronger than $s$: for this reason, if applicable, $r$ will defeat $s$. Suppose now to drop $r$. Nothing is said about the relation bewteen obligations and intentions, and so about rules $s$ and $t$. This means that there is no incompatibility relation between INT and OBL and we are free to derive both INT$q$ and OBL$\neg q$.

The relation Conflict is explicitly linked to that of agent type. Classically, agent types are characterised by stating conflict resolution types in terms of orders of overruling between rules [4,5,7,6]. In this perspective, agent types are meaningful within a non-monotonic setting and are nothing but general strategies to detect and solve conflicts between the different components of the cognitive profiles of agent's deliberation. In [4] 24 possible types are identified while, in [6], based on a different framework, 20 combinations are proposed. Typically, rational agents are assumed to be at least *realistic*: a realistic agent, in fact, is such that rules for beliefs override all other components, as beliefs correspond to agent's account of how the environment is. If the realistic condition is abandoned, we may have situations where intentions and desires override beliefs, thus leading to various forms of wishful thinking. Given the minimal assumption that a rational agent should be realistic, we may further constrain agent's deliberation in order not to violate obligations: a *social agent* type requires that obligations are stronger than the other motivational components with the exception of beliefs. Other agent types can be specified, for which see Section 4.

### 3.3 The Language of Modal Defeasible Logic

The inference process derives factual knowledge (through belief rules), intentions and obligations based on existing facts, intentions and obligations. Thus, rules allow for the derivation of new motivational factors of an agent. As was mentioned, we divide the rules into rules for beliefs, intentions, and obligations. Provability for beliefs does not generate modalised literals, since in our view beliefs concern the knowledge an agent has about the world and corresponds to the basic infernce mechanism of the agent.

A defeasible agent theory consists of a set of *facts* or indisputable statements, three sets of rules for beliefs, intentions, and obligations, a set of *conversions*

saying when a rule of one type can be used also as another type, a set of *conflict relations* saying when two rule types can be in conflict and which rule type prevails, and a *superiority relation* $>$ among rules saying when a single rule may override the conclusion of another rule. For $X \in \{\mathrm{BEL}, \mathrm{INT}, \mathrm{OBL}\}$, we have that $\phi_1, \ldots, \phi_n \rightarrow_X \psi$ is a *strict rule* such that whenever the premises $\phi_1, \ldots, \phi_n$ are indisputable so is the conclusion $\psi$. $\phi_1, \ldots, \phi_n \Rightarrow_X \psi$ is a *defeasible rule* that can be defeated by contrary evidence. $\phi_1, \ldots, \phi_n \rightsquigarrow_X \psi$ is a *defeater* that is used to defeat some defeasible rules by producing evidence to the contrary. It is worth noting that modalised literals can occur only in the antecedent of rules: the reason of this is that the rules are used to derive modalised conclusions while we do not conceptually need to iterate modalities. This limitation makes the system more manageable.

**Definition 1 (Language).** *Let* PROP *be a set of propositional atoms,* $\mathrm{MOD} = \{\mathrm{BEL}, \mathrm{INT}, \mathrm{OBL}\}$ *be the set of modal operators, and* Lab *be a set of labels. The sets below are the smallest sets closed under the following rules:*

***Literals***

$$\mathrm{Lit} = \mathrm{PROP} \cup \{\neg p | p \in \mathrm{PROP}\}$$

*If $q$ is a literal, $\sim q$ denotes the complementary literal (if $q$ is a positive literal $p$ then $\sim q$ is $\neg p$; and if $q$ is $\neg p$, then $\sim q$ is $p$);*

***Modal literals***

$$\mathrm{ModLit} = \{Xl, \neg Xl | l \in \mathrm{Lit}, X \in \{\mathrm{INT}, \mathrm{OBL}\}\};$$

***Rules*** $\mathrm{Rule} = \mathrm{Rule}_s \cup \mathrm{Rule}_d \cup \mathrm{Rule}_{dft}$, *where for* $X \in \mathrm{MOD}$

$$\begin{aligned}
\mathrm{Rule}_s &= \{r : \phi_1, \ldots, \phi_n \rightarrow_X \psi | \\
&\qquad r \in \mathrm{Lab}, A(r) \subseteq \mathrm{Lit} \cup \mathrm{ModLit}, \psi \in \mathrm{Lit}\} \\
\mathrm{Rule}_d &= \{r : \phi_1, \ldots, \phi_n \Rightarrow_X \psi | \\
&\qquad r \in \mathrm{Lab}, A(r) \subseteq \mathrm{Lit} \cup \mathrm{ModLit}, \psi \in \mathrm{Lit}\} \\
\mathrm{Rule}_{dft} &= \{r : \phi \rightsquigarrow_X \psi | \\
&\qquad r \in \mathrm{Lab}, A(r) \subseteq \mathrm{Lit} \cup \mathrm{ModLit}, \psi \in \mathrm{Lit}\}
\end{aligned}$$

*We use some obvious abbreviations, such as superscript for mental attitude, subscript for type of rule, and* $\mathrm{Rule}[\phi]$ *for rules whose consequent is $\phi$, for example:*

$$\begin{aligned}
\mathrm{Rule}^{\mathrm{BEL}} &= \{r : \phi_1, \ldots, \phi_n \rhd_{\mathrm{BEL}} \psi | \\
&\qquad (r : \phi_1, \ldots, \phi_n \rhd_{\mathrm{BEL}} \psi) \in \mathrm{Rule}, \rhd \in \{\rightarrow, \Rightarrow, \rightsquigarrow\}\} \\
\mathrm{Rule}_s[\psi] &= \{\phi_1, \ldots, \phi_n \rightarrow_X \psi | \\
&\qquad \{\phi_1, \ldots, \phi_n\} \subseteq \mathrm{Lit} \cup \mathrm{ModLit}, \psi \in \mathrm{Lit}, X \in \mathrm{MOD}\}
\end{aligned}$$

*We use $A(r)$ to denote the set $\{\phi_1, \ldots, \phi_n\}$ of* antecedents *of the rule $r$, and $C(r)$ to denote the* consequent *$\psi$ of the rule $r$.*

**Definition 2 (Conversion and Conflict Relations).** *The conversion relation* Convert *is defined as follows:*

$$\text{Convert} \subseteq \text{MOD} \times \text{MOD}$$

*The conflict relation* $\text{Conflict} \subseteq \text{MOD} \times \text{MOD}$ *is such that*

$$\forall X, Y \in \text{MOD},\ \text{Conflict}(X, Y) \Rightarrow \neg(\text{Conflict}(Y, X))\ \textit{(asymmetry)}$$

**Definition 3 (Defeasible Agent Theory).** *A defeasible agent theory is a structure*

$$D = (F, R^{\text{BEL}}, R^{\text{INT}}, R^{\text{OBL}}, >, \mathcal{C}, \mathcal{V})$$

*where*

- $F \subseteq \text{Lit} \cup \text{ModLit}$ *is a finite set of facts;*
- $R^{\text{BEL}} \subseteq \text{Rule}^{\text{BEL}}$, $R^{\text{INT}} \subseteq \text{Rule}^{\text{INT}}$, $R^{\text{OBL}} \subseteq \text{Rule}^{\text{OBL}}$ *are three finite sets of rules such that each rule has a unique label;*
- *The superiority relation* $>$ *is such that* $>=>^{st} \cup >^{\text{Conflict}}$, *where* $>^{st} \subseteq R^X \times R^X$ *such that if* $r > s$, *then if* $r \in \text{Rule}^X[p]$ *then* $s \in \text{Rule}^X[\sim p]$ *and* $>$ *is acyclic; and* $>^{\text{Conflict}}$ *is such that*

$$\forall r \in \text{Rule}^X[p], \forall s \in \text{Rule}^Y[\sim p],\ \textit{if}\ \text{Conflict}(X, Y),\ \textit{then}\ r >^{\text{Conflict}} s$$

- $\mathcal{C} \subseteq \{\text{Convert}(X, Y) | X, Y \in \text{MOD}\}$ *is a set of conversions;*
- $\mathcal{V} \subseteq \{\text{Conflict}(X, Y) | X, Y \in \text{MOD}\}$ *is a set of conflict relations.*

The following running example illustrates the defeasible agent theory.

**Example 1** (Running example). Frodo, our Tolkienian agent, is entrusted by Elrond to be the bearer of the ring of power, a ring forged by the dark lord Sauron. Frodo has the task to bring the ring to Mordor, the realm of Sauron, and to destroy it by throwing it into the fires of Mount Doom. However, Frodo loves the place where he was born, the Shire, and intends to go there.

$$\begin{aligned}
F &= \{\text{INT}\,\textit{GoToShire},\ \textit{EntrustedByElrond}\} \\
R &= \{r_1 : \textit{EntrustedByElrond} \Rightarrow_{\text{BEL}} \textit{RingBearer} \\
&\qquad r_2 : \textit{RingBearer} \Rightarrow_{\text{OBL}} \textit{DestroyRing} \\
&\qquad r_3 : \text{INT}\,\textit{GoToShire} \Rightarrow_{\text{INT}} \neg \textit{GoToMordor} \\
&\qquad r_4 : \neg \textit{GoToMordor} \Rightarrow_{\text{BEL}} \neg \textit{DestroyRing}\} \\
>= &\{r_4 > r_2\} \\
\mathcal{C} &= \{\text{Convert}(\text{BEL}, \text{INT})\} \\
\mathcal{V} &= \{\text{Conflict}(\text{BEL}, \text{OBL})\}
\end{aligned}$$

### 3.4 Inferences with BIO Agents

Proofs are sequences of literals and modal literals together with so-called proof tags $+\Delta$, $-\Delta$, $+\partial$ and $-\partial$. Given a defeasible agent theory $D$, $+\Delta_X q$ means that literal $q$ is provable in $D$ using only facts and strict rules for modality $X$, $-\Delta_X q$ means that it has been proved in $D$ that $q$ is not definitely provable in $D$, $+\partial_X q$ means that $q$ is defeasibly provable in $D$, and $-\partial_X q$ means that it has been proved in $D$ that $q$ is not defeasibly provable in $D$.

**Definition 4.** *Given an agent theory $D$, a proof in $D$ is a linear derivation, i.e, a sequence of labelled formulas of the type $+\Delta_X q$, $-\Delta_X q$, $+\partial_X q$ and $-\partial_X q$, where the proof conditions defined in the rest of this section hold.*

We start with some terminology. As was explained, the following definition states the special status of belief rules, and that the introduction of a modal operator corresponds to being able to derive the associated literal using the rules for the modal operator.

**Definition 5.** *Let $\# \in \{\Delta, \partial\}$, and $P = (P(1), \ldots, P(n))$ be a proof in $D$. A (modal) literal $q$ is* #-provable *in $P$ if there is a line $P(m)$ of $P$ such that either*

1. *$q$ is a literal and $P(m) = +\#_{\mathrm{BEL}} q$ or*
2. *$q$ is a modal literal $Xp$ and $P(m) = +\#_X p$ or*
3. *$q$ is a modal literal $\neg Xp$ and $P(m) = -\#_X p$.*

*A literal $q$ is* #-rejected *in $P$ if there is a line $P(m)$ of $P$ such that*

1. *$q$ is a literal and $P(m) = -\#_{\mathrm{BEL}} q$ or*
2. *$q$ is a modal literal $Xp$ and $P(m) = -\#_X p$ or*
3. *$q$ is a modal literal $\neg Xp$ and $P(m) = +\#_X p$.*

The definition of $\Delta_X$ describes just forward chaining of strict rules:

$+\Delta_X$: If $P(n+1) = +\Delta_X q$ then
(1) $q \in F$ if $X = \mathrm{BEL}$ or $Xq \in F$ or
(2) $\exists r \in R_s^X[q] : \forall a \in A(r)$ $a$ is $\Delta$-provable or
(3) $\exists r \in R_s^Y[q] : \mathrm{Convert}(Y, X) \in \mathcal{C}$, $\forall a \in A(r)$ $Xa$ is $\Delta$-provable.

$-\Delta_X$: If $P(n+1) = -\Delta_X q$ then
(1) $q \notin F$ if $X = \mathrm{BEL}$ and $Xq \notin F$ and
(2) $\forall r \in R_s^X[q]$ $\exists a \in A(r) : a$ is $\Delta$-rejected and
(3) $\forall r \in R_s^Y[q]$ : if $\mathrm{Convert}(Y, X) \in \mathcal{C}$ then $\exists a \in A(r)$ $Xa$ is $\Delta$-rejected.

For a literal $q$ to be definitely provable with the mode $X$ we need to find a strict rule for $X$ with head $q$, whose antecedents have all been definitely proved previously. And to establish that $q$ cannot be definitely proven we must establish that for every strict rule with head $q$ there is at least one antecedent which has been shown to be non-provable. Condition (3) says that a rule for $Y$ can be used as a rule for a different modal operator $X$ in case all literals in the body of the rule are modalised with the modal operator we want to prove. For example,

given the rule $p, q \rightarrow_{\text{BEL}} s$, we can derive $+\Delta_{\text{INT}} s$ if we have $+\Delta_{\text{INT}} p$, $+\Delta_{\text{INT}} q$, and the conversion Convert(BEL, INT) holds in the theory.

Conditions for $\partial_X$ are more complicated. We define when a rule is applicable or discarded. A rule for a belief is applicable if all the literals in the antecedent of the rule are provable with the appropriate modalities, while the rule is discarded if at least one of the literals in the antecedent is not provable. As before, for the other types of rules we have to take conversions into account. We have thus to determine conditions under which a rule for $Y$ can be used to directly derive a literal $q$ modalised by $X$. Roughly, the condition is that all the antecedents $a$ of the rule are such that $+\partial_X a$.

**Definition 6.** *Given a derivation $P$, $P(1..n)$ denotes the initial part of the derivation of length $n$. Let $X, Y, Z \in$ MOD.*

- *A rule $r \in R_{sd}$ is* applicable *in the proof condition for $\pm\partial_X$ iff*
  1. *$r \in R^X$ and $\forall a \in A(r)$, $+\partial_{\text{BEL}} a \in P(1..n)$ and $\forall Z a \in A(r)$, $+\partial_Z a \in P(1..n)$, or*
  2. *$r \in R^Y$, $\text{Convert}(Y, X) \in \mathcal{C}$, and $\forall a \in A(r)$, $+\partial_X a \in P(1..n)$.*
- *A rule $r$ is* discarded *in the condition for $\pm\partial_X$ iff*
  1. *$r \in R^X$ and $\exists a \in A(r)$ such that $-\partial_{\text{BEL}} a \in P(1..n)$ or $\exists Z a \in A(r)$ such that $-\partial_Z a \in P(1..n)$; or*
  2. *$r \in R^Y$ and, if $\text{Convert}(Y, X)$, then $\exists a \in A(r)$ such that $-\partial_X a \in P(1..n)$, or*
  3. *$r \in R^Z$ and either $\neg\text{Convert}(Z, X)$ or $\neg\text{Conflict}(Z, X)$.*

**Example 2** The rule $a, \text{INT}b \Rightarrow_{\text{BEL}} c$ is applicable if we can prove both $+\partial_{\text{BEL}} a$ and $+\partial_{\text{INT}} b$.

**Example 3** If we have a type of agent that allows a deontic rule to be converted into a rule for intention, Convert(OBL, INT), then the definition of applicable in the condition for $\pm\partial_{\text{INT}}$ is as follows: a rule $r \in R_{sd}[q]$ is applicable iff (1) $r \in R^{\text{INT}}$ and $\forall a \in A(r)$, $+\partial_{\text{BEL}} a \in P(1..n)$ and $\forall X a \in A(r)$, $+\partial_X a \in P(1..n)$, (2) or $r \in R^{\text{OBL}}$ and $\forall a \in A(r)$, $+\partial_{\text{INT}} a \in P(1..n)$. In this second case, for example, given the rule $p, q \Rightarrow_{\text{OBL}} s$, we can derive $+\partial_{\text{INT}} s$ if we have $+\partial_{\text{INT}} p$ and $+\partial_{\text{INT}} q$.

As a corollary of the definition of applicability, we can establish when a literal is supported (see Section 5.2 for the use of this notion):

**Definition 7.** *Given a theory $D$, a literal $l$ is* supported *in $D$ iff there exists a rule $r \in R[l]$ such that $r$ is applicable, otherwise $l$ is not supported. For $X \in$ MOD we use $+\Sigma_X l$ and $-\Sigma_X l$ to indicate that $l$ is supported / not supported by rules for $X$.*

We are now ready to provide proof conditions for $\pm\partial_X$:

$+\partial_X$: If $P(n+1) = +\partial_X q$ then
(1)$+\Delta_X q \in P(1..n)$ or
(2) (2.1) $-\Delta_X {\sim} q \in P(1..n)$ and
(2.2) $\exists r \in R_{sd}[q]$ such that $r$ is applicable, and
(2.3) $\forall s \in R[{\sim} q]$ either $s$ is discarded, or
(2.3.1) $\exists t \in R[q]$ such that $t$ is applicable and $t > s$, and either
$t, s \in R^Z$, or $\mathrm{Convert}(Y, X)$ and $r \in R^Y$

$-\partial_X$: If $P(n+1) = -\partial_X q$ then
(1) $-\Delta_X q \in P(1..n)$ and either
(2.1) $+\Delta_X {\sim} q \in P(1..n)$ or
(2.2) $\forall r \in R_{sd}[q]$, either $r$ is discarded, or
(2.3) $\exists s \in R[{\sim} q]$, such that $s$ is applicable, and
(2.3.1) $\forall t \in R[q]$ either $t$ is discarded, or $t \not> s$, or
$t, s \notin R^Z$ and, if $r \in R^Y$ then $\neg\mathrm{Convert}(Y, X)$.

To show that $q$ is defeasibly provable we have two choices: (1) We show that $q$ is already definitely provable; or (2) we need to argue using the defeasible part of a theory $D$. For this second case, three (sub)conditions must be satisfied. First, we require that there must be a strict or defeasible rule for $q$ which can be applied (2.1). Second, we need to consider possible reasoning chains in support of ${\sim}q$, and show that ${\sim}q$ is not definitely provable (2.2). Third, we must consider the set of all rules which are not known to be inapplicable and which permit to get ${\sim}q$ (2.3). Essentially, each such a rule $s$ attacks the conclusion $q$. For $q$ to be provable, $s$ must be counterattacked by a rule $t$ for $q$ with the following properties: (i) $t$ must be applicable, and (ii) $t$ must be stronger than $s$. Thus each attack on the conclusion $q$ must be counterattacked by a stronger rule. In other words, $r$ and the rules $t$ form a team (for $q$) that defeats the rules $s$. $-\partial_X q$ is defined in an analogous manner.

**Example 4** (Running example; continued). Below is the set $C$ of all conclusions we get using the rules in $R$:

$$C = \{\mathit{RingBearer},\ \mathrm{INT}\neg\mathit{GoToMordor},\ \mathrm{INT}\neg\mathit{DestroyRing}\}$$

As facts, we know that Frodo has the primitive intention to go to the Shire and that he has been entrusted by Elrond. These facts make applicable rules $r_3$ and $r_1$, which permit to derive that Frodo is the ring bearer and that he has the intention not to go to Mordor. At this point we have a conflict, as we have $\mathrm{Conflict}(\mathrm{BEL}, \mathrm{OBL})$ and $\mathrm{Convert}(\mathrm{BEL}, \mathrm{INT})$. In effect, given the conversion, $r_4$ permits to derive that Frodo has the intention not to destroy the ring while rule $r_2$ should lead to the obligation to destroy it. However, $r_4$ is stronger than $r_2$ and so we only get $+\partial_{\mathrm{INT}}\neg\mathit{DestroyRing}$.

## 4 Agent Types

Classically, agent types are characterized by stating conflict resolution types in terms of orders of overruling between rules [4,5]. For example, an agent is *realistic*

when rules for beliefs override all other components; she is *social* when obligations are stronger than the other motivational components with the exception of beliefs, etc.

As suggested in [7,6], agent types can be characterised in DL as follows:

**Definition 8 (Agent Type (1)).** *An agent type is defined by a set of pairs* $(X, Y)$*,* $X, Y \in \{\text{BEL}, \text{OBL}, \text{INT}\}$*, such that for every* $r$ *and* $r'$ *such that* $r \in R^X[q]$ *and* $r' \in R^Y[\sim q]$*, we have that* $r > r'$*.*

For example, while realistic agents are such that $X = \text{BEL}$ and $Y \in \{\text{INT}, \text{OBL}\}$, social agents are such that $X = \text{OBL}$ and $Y = \text{INT}$. It is clear that the notion of agent type is defined in terms of the relation Conflict we have previously introduced.

Let us see the agent types that can be identified in the framework we have defined so far. Table 1 shows all possible cases and, for each kind of rule, indicates all attacks on it. It should be read as follows. Each of the three main columns identifies a possible kind of conflict between two types $X, Y$ of applicable rules that would permit to infer the literals $p$ and $\sim p$ labelled by $X$ and $Y$ respectively. The first row from the top in the three main columns specifies the case where both literals are derived (i.e., there is no conflict, which indeed corresponds to the case where the modalities involved are not in Conflict); the second and third rows from top identify the cases where we have a conflict and one rule prevails over the other. The third sub-column in each main column defines the agent type for which each conflict-detection and -resolution policy is appropriate. (To save space, in Table 1 "indep." abbreviates "independent", "wish. th." "wishful thinking", and "real." "realistic".)

| $\Rightarrow_{\text{OBL}} p$ / $\Rightarrow_{\text{INT}} \sim p$ | | | $\Rightarrow_{\text{OBL}} p$ / $\Rightarrow_{\text{BEL}} \sim p$ | | | $\Rightarrow_{\text{INT}} p$ / $\Rightarrow_{\text{BEL}} \sim p$ | | |
|---|---|---|---|---|---|---|---|---|
| $+\partial_{\text{OBL}} p$ | $+\partial_{\text{INT}} \sim p$ | indep. | $+\partial_{\text{OBL}} p$ | $+\partial_{\text{BEL}} \sim p$ | wish. th. | $+\partial_{\text{INT}} p$ | $+\partial_{\text{BEL}} \sim p$ | wish. th. |
| $+\partial_{\text{OBL}} p$ | $-\partial_{\text{INT}} \sim p$ | social | $+\partial_{\text{OBL}} p$ | $-\partial_{\text{BEL}} \sim p$ | wish. th. | $+\partial_{\text{INT}} p$ | $-\partial_{\text{BEL}} p$ | wish. th. |
| $-\partial_{\text{OBL}} p$ | $+\partial_{\text{INT}} \sim p$ | deviant | $-\partial_{\text{OBL}} p$ | $+\partial_{\text{BEL}} \sim p$ | real. | $-\partial_{\text{INT}} p$ | $+\partial_{\text{BEL}} \sim p$ | real. |

**Table 1.** Conflict: Agent Types

Independent agents are free to adopt intentions for $p$ in presence of derivations for OBL$\sim p$. This is possible in our framework when we have that $\neg\text{Conflict}(\text{OBL}, \text{INT})$ and $\neg\text{Conflict}(\text{INT}, \text{OBL})$: this means that the system admits both conclusions, as they are not in conflict. As expected, for social agents obligations override intentions and so $\text{Conflict}(\text{OBL}, \text{INT})$; the opposite case is when an agent is deviant and her intentions override the obligations, $\text{Conflict}(\text{INT}, \text{OBL})$. Where beliefs are defeated either by obligations or by intentions we have classical examples of wishful thinking. Notice that in Table 1 also the cases $+\partial_{\text{OBL}} p / +\partial_{\text{BEL}} \sim p$ and $+\partial_{\text{INT}} p / +\partial_{\text{BEL}} \sim p$ have been classified as wishful thinking, given the basic nature of beliefs we adopted in our framework. However, we are aware that this reading is debatable: in effect, if we can derive

both conclusions, this means that there is no real conflict. Last, it is worth noting that we do not consider here the case where $-\partial_X p / -\partial_Y \sim p$: here we would have that $X$ and $Y$ are incompatible, but that it is not possible to establish what rule is the strongest one, thus leading to a mutual defeating of the rules involved. This case –which is discussed in [5,7,6] and permits to identify other agent types– is excluded here, as the relation Conflict both identifies conflicts and solves them by establishing what rule type must prevail.

It is possible to integrate the above classifications by referring to the notion of conversion. Conversions do not have a direct relation with conflict resolution because they simply affect the condition of applicability of rules. However, they indeed contribute to define the cognitive profile of agents because they allow to obtain conclusions modalised by a certain $X$ through the application of rules which are not modalised by $X$. Table 2 shows the conversions and specify new agent types with respect to which each conversion seems to be appropriate.

| | | | |
|---|---|---|---|
| Convert(BEL, OBL) | c-realistic | Convert(INT, OBL) | c-deviant |
| Convert(BEL, INT) | c-realistic | Convert(OBL, BEL) | NO |
| Convert(OBL, INT) | c-social | Convert(INT, BEL) | NO |

**Table 2.** Conversions

A preliminary remark before commenting Table 2. We do not consider here conversions Convert$(X, Y)$ where $X = Y$. In fact, even though they can be admitted, they do not seem to characterise a specific cognitive profile for the agents. Consider Convert(BEL, OBL) and Convert(BEL, INT). Both seem appropriate for some types of realistic agent. Indeed, for a realistic agent beliefs correspond to her basic reasoning mechanism. Accordingly, if we have

$$r : \neg \mathit{open\_umbrella} \Rightarrow_{\mathrm{BEL}} \mathit{wet}$$

$$+\partial_{\mathrm{INT}} \neg \mathit{open\_umbrella} \qquad +\partial_{\mathrm{OBL}} \neg \mathit{open\_umbrella}$$

it is reasonable to derive both that the agent has the intention to be wet, and that it is obligatory for her to be wet.

Other conversions look more appropriate for other agent types. For example, we may have agent types for which Convert(OBL, INT) holds. This means that from

$$r : \mathit{kill} \Rightarrow_{\mathrm{OBL}} \mathit{kill\_gently} \; +\partial_{\mathrm{INT}} \mathit{kill}$$

we can derive that the agent has the intention to kill gently. But this derivation is conceptually meaningful only if we assume a kind of norm regimentation, by which we impose that all agents intend what is prescribed by deontic rules.

The peculiarity of Convert(INT, OBL) is that the simple fact that something is derived as obligatory can permit to obtain through a rule for intention that something else is obligatory as well. Consider this case:

$$r : \mathit{help\_needy\_people} \Rightarrow_{\mathrm{INT}} \mathit{save\_money} \; +\partial_{\mathrm{OBL}} \mathit{help\_needy\_people}$$

If Convert(INT, OBL) holds, then we can derive that it is obligatory for the agent to save money: an intention supports the derivation of an obligation. In other papers [5], this case has been classified as an example of an agent legislator. Here, we prefer to consider it as a case of a deviant agent [6], due to its structural similarity to Conflict(INT, OBL).

Finally, notice that the conversions Convert(OBL, BEL) and Convert(INT, BEL), which are marked in the table by a "NO", seem meaningless. They say that a rule for obligation and for intention may respectively be used to derive a belief. This sounds odd, at least adopting the interpretation of beliefs of this paper. In fact, since the belief modality captures the basic knoweledge the agent has about the environment, it is treated as its logic were reflexive (namely, that BEL$\psi \rightarrow \psi$ holds). Consider, for example, the following:

$$r : \mathit{help\_needy\_people} \Rightarrow_{\mathrm{INT}} \mathit{save\_money} + \partial_{\mathrm{BEL}} \mathit{help\_needy\_people}$$

If Convert(INT, BEL) holds, then we obtain that the agent in fact saves money, which is odd: beliefs, according to our interpretation should be independent from agent's deliberation, even though they are used to derive motivational attitudes such as intentions and obligations. In addition, adopting both Convert(OBL, BEL) and Convert(INT, BEL) would determine a collapse of our logic, as we could dispense with explicit modalities in the antecedent of rules.

Since our logic system is characterised by Conflict as well as by Convert, and conversions indeed contribute to define the cognitive profile of an agent, it seems that an agent type should take both parameters into account:

**Definition 9 (Agent Type (2)).** *An agent type is defined by a pair* $(\Gamma, \Delta)$*, where* $\Gamma \subseteq \{\text{Conflict}(X, Y) | X, Y \in \text{MOD}\}$ *and* $\Delta \subseteq \{\text{Convert}(Z, W) | Z, W \in \text{MOD}\}$.

It is easy to see that the notion of agent type of Definition 8 (proposed in [7,6]) is captured by Definition 9.

This completes our picture of the notion of agent types. However, a serious difficulty is around the corner when we focus on the notion of agent type based on defining criteria for conflict-detection and -resolution. Are we sure that this view is sufficient, given the account of policy-based attitudes we previously discussed? In the reminder we will consider only the interaction between intentions and obligations, event though similar remarks can be easily extended to all other agent types presented in Table 1. But, even confining the problem to these components, the question at stake is: How to deal with social agents? The simplest solution is the classical one, corresponding to adopting schema (5) and that we have adopted so far: when we have two rules, one leading to INT$\phi$ and the other to OBL$\neg\phi$, the former is blocked. As we shall see, this strategy is not enough.

## 5 Social Agents

### 5.1 The Problem

The idea of social agent based on the intuition of Definition 8–which is also adopted in [4]– does not guarantee that agent's deliberation is oriented to fully

complying with obligations. The same holds when Definition 9 is used. In effect, to our view a social agent can be defined by the following pair

$$(\{\text{Conflict}(\text{BEL}, \text{INT}), \text{Conflict}(\text{BEL}, \text{OBL}), \text{Conflict}(\text{OBL}, \text{INT})\},$$
$$\{\text{Convert}(\text{BEL}, \text{OBL}), \text{Convert}(\text{BEL}, \text{INT})\})$$

according to which the agent is realistic (beliefs override the other components, and the appropriate conversions hold) and obligations prevail over conflicting intentions.

In both cases, the drawback is mainly due to the introduction of conversions. Since conversions allow to obtain conclusions modalised by a certain $X$ through the application of rules which are not modalised by $X$, they are fundamental in order to capture the fact that some side-effects should be accepted insofar as they are consequences of policies of which the agent is aware. Moreover, some conversions seem useful to integrate the basic idea of social agency.

It is clear that our system admits three different types of intentions and obligations. First, we have *primitive* intentions and obligations when these are facts of the theory. But we can also have what we may call *primary* and *secondary* intentions and obligations, depending on whether we accept at least basic conversions via belief rules.

Let us consider Example 1. INT*GoToShire* is a primitive intention. On the other hand, OBL*DestroyRing* –if it were derived from rule $r_2$– and INT$\neg$*GoToMordor* are primary obligations and intentions as they would be obtained without the use of conversions (see Example 4). Finally, INT$\neg$*DestroyRing* is a secondary intention because it is obtained from the rule $r_4 : \neg GoToMordor \Rightarrow_{\text{BEL}} \neg DestroyRing$ and from $+\partial_{\text{INT}} \neg GoToMordor$ (again, see Example 4). It should be noted that OBL*DestroyRing* cannot be derived because $r_4 > r_2$, but this just amounts to assuming that the agent is realistic: $r_4$ is a belief rule whereas $r_2$ is a deontic rule. In other words, when we have in general that

$$\begin{array}{ll} a \Rightarrow_{\text{OBL}} q & b \Rightarrow_{\text{BEL}} \sim q \\ +\partial_{\text{BEL}} a & +\partial_{\text{INT}} b \end{array}$$

we are doomed to have social agents who cannot be truly social since some of their (primitive) intentions lead to behaviours against what would be otherwise obligatory for the agents. However, this issue is not a matter of a direct conflict between rules for intentions and obligations. Thus, to deal with norm-complying agents in these scenarios and to restore their sociality we are required to change the notion of agent type. We cannot anymore define it in terms of an order of overruling between rules, but we have to focus on how the conflicting literals are derived during the proof. Indeed, this is feasible, but has a high computational cost, and even then we cannot guarantee the sociality of an agent.

### 5.2 The Cost of Social Agents

In this section we investigate the complexity of the defeasible logic for BIO agents where we assume the conversions Convert(BEL, OBL) and Convert(BEL, INT)

and then we turn our attention to the complexity of social agents. We first introduce some notions to make precise the definition of the issues at hand.

**Definition 10.** *Let # be one of the proof tags. Given a theory $D$, $D \vdash \pm\#p$ iff there is a derivation $P$ in $D$ such that for some $n$ $P(n) = \pm\#p$.*

**Definition 11.** *Given a theory $D$, the* universe *of $D$ ($U^D$) is the set of all the atoms occurring in $D$; the* extension *of $D$ ($E^D$), is defined as follows:*

$$E^D = (\Delta^+, \Delta^-, \partial^+, \partial^-)$$

*where for $X \in \{\text{BEL}, \text{INT}, \text{OBL}\}$*

$$\begin{aligned}
\Delta^+ &= \{Xl : D \vdash +\Delta_X l\};\\
\Delta^- &= \{Xl : D \vdash -\Delta_X l\};\\
\partial^+ &= \{Xl : D \vdash +\partial_X l\};\\
\partial^- &= \{Xl : D \vdash -\partial_X l\}.
\end{aligned}$$

Two theories $D$ and $D'$ are *equivalent* if and only if they have the same extension, namely $D \equiv D'$ iff $E^D = E^{D'}$.

We now prove the main theorem about the complexity of our defeasible logic. We show that the logic has linear complexity if we compute the whole set of conclusions, i.e., the extension, of a given theory.

**Theorem 1.** *For every theory $D$, $E^D$ can be computed in time linear to the size of the theory, i.e., $O(|U^D| * |R|)$.*

*Proof.* The proof is based on a modification of the algorithm given by Maher [26] to show that propositional defeasible logic has linear complexity.

The main idea of the proof is to build appropriate data structure to implement a series of transformations reducing the complexity of the rules, and where each literal and modal literal is examined only once. The focal point of the transformations is based on the following properties:

- Let $D \vdash +\partial p$ then

$$D \cup \{r : p_1, \ldots, p_n, p \Rightarrow q\} \equiv D \cup \{r : p_1, \ldots, p_n \Rightarrow q\}.$$

- Let $D \vdash -\partial p$ then $D \cup \{r : p_1, \ldots, p_n, p \Rightarrow q\} \equiv D$.

The properties allow us (1) to remove already proved literals from the body of rules and (2) to remove rules which have been discarded.

The algorithm has three phases. (1) A pre-processing phase where we use similar transformations to those given in [24] to transform a theory into an equivalent theory without superiority relation and defeaters; the transformation is linear. (2) A *rule loader* that parses the theory obtained in the first phase and generates the data structure that encodes the theory. (3) The *inference engine* applies transformations to the data structure, where at every step it reduces the complexity of the data structure.

*(1) Transformations* Theory transformations are an important tools to study properties of defeasibly logic. In [24] we extensively used transformations to show under which conditions it is possible to simplify the presentation of basic defeasible logic by dispensing defeaters and the superiority relation. In what follows we are going to give transformations that allow us to remove defeaters and the superiority relation from modal defeasible theories for BIO agents.

**Definition 12.** *Let # two modal defeasible theories $D_1$ and $D_2$ are equivalent (written $D_1 \equiv D_2$) iff $\forall p, D_1 \vdash \#p$ iff $D_2 \vdash \#p$, i.e., they have the same consequences. Similarly $D_1 \equiv_\Sigma D_2$ means that $D_1$ and $D_2$ have the same consequences in the language $\Sigma$.*

**Definition 13.** *A* transformation *is a mapping from modal defeasible theories to modal defeasible theories. A transformation $T$ is* correct *iff for all modal defeasible theories $D$, $D \equiv_\Sigma T(D)$ where $\Sigma$ is the language of $D$.*

**Definition 14.** *Let $A = \{l_1, \ldots, l_n\} \subseteq$ Lit and $X \in$ MOD, then $XA = \{Xl_i : l_i \in A\}$.*

**Definition 15.** *Let $D = (F, R, >)$ be a defeasible theory such that $R_{dft} = \emptyset$. Let $\Sigma$ be the language of $D$. Define $elimsup(D) = (F, R', \emptyset)$, where*

$$R' = \{\neg inf(r) \Rightarrow_{\mathrm{BEL}} inf(s) : (r, s) \in >\} \bigcup_{r \in R} elimsup(r)$$

*and*

$$elimsup(r) = \{A(r) \hookrightarrow_{\mathrm{BEL}} \neg inf(r), \neg inf(r) \hookrightarrow_X C(r) : A(r) \hookrightarrow_X C(r) \in R_{sd}\}$$

*For each rule $r \in R$, $inf(r)$ is a new atom, i.e., they do not appear in $\Sigma$. Furthermore all new atoms generated are distinct.*

**Theorem 2.** *The transformation elimsup is correct.*

*Proof.* The proof by induction on the length of derivations is similar to that given in [24]. Here we give in full the case of strict derivations and we outline the main part of the case of defeasible derivations.

Case if $D \vdash +\Delta_X p$ then $elimsup(D) \vdash +\Delta_X p$. For a proof of length 1 of $+\Delta_X p$, i.e., $P(1) = +\Delta_X p$, then we have two cases: (1) $Xp \in F$, (2) $\exists r \in R_s^X[p], A(r) = \emptyset$. The first case is trivial since $F$ is the same in $D$ and $elimsup(D)$. For (2) we have that $elimsup(D)$ contains the rules $r^a :\rightarrow_{\mathrm{BEL}} \neg inf(r)$, and $r^c : \neg inf(r) \rightarrow p$. $r^a$ is applicable, so we have $+\Delta_{\mathrm{BEL}} \neg inf(r)$, then this makes $r^c$ applicable and then we have $elimsup(D) \vdash +\Delta_X p$.

For the inductive step, we assume as usual that the property holds for proofs whose length is up to $n$, and than we consider $P(n+1) = +\Delta_X p$. Beside the cases for the inductive base, we have two additional cases to consider here: (a) $\exists r \in R_s^X[p], \forall a \in A(r), +\Delta_Y a \in P(1..n)$; (b) Convert$(X, Y)$ and $\exists s \in R_s^Y, +\Delta_X a \in P(1..n)$.

For (a) by inductive hypothesis, $\forall a \in A(r)$, $elimsup(D) \vdash +\Delta a$, thus the rule $r^a : A(r) \rightarrow_{\text{BEL}} \neg inf(r)$ is applicable, thus $elimsup(D) \vdash +\Delta_{\text{BEL}} \neg inf(r)$, which makes rule $r^c : \neg inf(r) \rightarrow_X p$ applicable as well, and we can conclude $elimsup(D) \vdash +\Delta_X p$.

For (b) by inductive hypothesis $\forall a \in A(r)$, $elimsup(D) \vdash +\Delta_X a$, thus we can use the rule $s^a : A(r) \rightarrow_{\text{BEL}} \neg inf(s)$ to derive $+\Delta_X \neg inf(s)$. Since we have Convert$(Y, X)$, we can apply conversion to the rule $s^c : \neg inf(s) \rightarrow_Y p$ to derive $elimsup(D) \vdash +\Delta_X p$.

For the other direction, i.e., $elimsup(D) \vdash +\Delta_X p$ (for $p \in \Sigma$) then $D \vdash +\Delta_X p$, the proof is again by induction on the length of derivations.

The inductive base is trivial since the only possible derivation for a modal literal $Xp$ in $\Sigma$ is only when $Xp \in F$, and thus $Xp$ is also a fact in $D$,

For the inductive bases, $P(n+1) = +\Delta_X p$ we have that for every literal in $\Sigma$ which is not a fact, $\exists r \in R$ such that either (i) $\neg inf(r) \rightarrow_X p$ or (ii) $\neg inf(r) \rightarrow_Y p$ is in $elimsup(D)$. In addition we have a rule $A(r) \rightarrow_{\text{BEL}} \neg inf(r)$.

For (i) in the proof we have $\forall a \in A(r), +\Delta a \in P(1..n)$, thus by inductive hypothesis $D \vdash +\Delta a$, which makes applicable the rule $r : A(r) \rightarrow_X p$. For (ii) to derive $+\Delta_X p$ from $\neg inf(r) \rightarrow_Y p$, we must have $+\Delta_X \neg inf(r) \in P(1..n)$, which means that we have $+\Delta_X a \in P(1..n)$ for all $a \in A(r)$. Again by inductive hypothesis we have $D \vdash +\Delta_X a$ for all $a \in A(r)$, and $r$ is $A(r) \rightarrow_Y p$ were Convert$(Y, X)$. Therefore $D \vdash +\Delta_X p$.

The proof for $-\Delta_X$ is analogous and uses the same ideas of conversion from the case for $+\Delta$ and the basic structure from the proof for the transformation that removes the superiority relation from [24].

The proof of the case for $+\partial$ is essentially the same as that given in [24]. The only difference is in the iterative construction of the sets of maximal applicable rules, the existence of such sets is guaranteed by the clause of the proof conditions saying $t > s$. If a rule $r$ is maximal applicable then either $\forall a \in A(r), +\partial a$ or $\forall a \in A(r), +\partial_X a$ (applicable condition), and there is no applicable rule $s$ such that $s > r$. Thus all rules $\neg inf(s) \hookrightarrow_{\text{BEL}} inf(r)$ are discarded while the rule $A(r) \hookrightarrow_{\text{BEL}} \neg inf(r)$ is applicable, thus we prove either $+\partial_{\text{BEL}} \neg inf(r)$ or $+\partial_X \neg inf(r)$. Thus every rule $\neg inf(r) \Rightarrow_{\text{BEL}} inf(t)$, is applicable, this means that we prove $-\partial_Z \neg inf(t)$ for all $Z \in$ MOD. The main points here is that BEL converts universally and that there are conflict between all pairs of modalities. Accordingly the rule $t^c$, attacking a rule for $p$ is discarded. Using all rules in the maximal applicable sets we can show that all rules attacking $p$ are discarded, and that we have at least one applicable rule for $p$. The proof for $-\partial$ has the same structure of that given in [24] for the same case and the construction just outlined for the case $+\partial$.

**Definition 16.** *Let $D = (F, R, > 0)$ be a modal defeasible theory, and $\Sigma$ be the language of $D$. Define $elimdft = (F, R', >')$ where*

$$R' = \bigcup_{r \in R} elimdft(r)$$

*and*

$$elimdft(r) = \begin{cases} \{r^+ : A(r) \hookrightarrow_{\mathrm{BEL}} p^+, r^- : A(r) \hookrightarrow_{\mathrm{BEL}} \neg p^-, & \\ \quad r : p^+ \hookrightarrow_X p\} & r \in R^X_{sd}[p] \\ \{r^- : A(r) \hookrightarrow_{\mathrm{BEL}} p^-, r^+ : A(r) \hookrightarrow_{\mathrm{BEL}} \neg p^+, & \\ \quad r : p^- \hookrightarrow_X \neg p\} & r \in R^X_{sd}[\neg p] \\ \{r : A(r) \Rightarrow_{\mathrm{BEL}} \neg p^-\} & r \in R_{dft}[p] \\ \{r : A(r) \Rightarrow_{\mathrm{BEL}} \neg p^+\} & r \in R_{dft}[\neg p] \end{cases}$$

*and the superiority relation $>'$ is defined by the following conditions:*

$$\forall r', s' \in R'(r' >' s' \iff \exists r, s \in R : r' \in elimdft(r), s' \in elimdft(s), r > s)$$

*where $r$ and $s$ are conflicting.*

*For each atom $p \in \Sigma$, $p^+$ and $p^-$ are new atoms, i.e., thy do not appear in $\Sigma$. Furthermore all new atoms generated are distinct.*

**Theorem 3.** *The transformation elimdft is correct.*

*Proof.* Notice that the transformation *elimdft* is essentially the same transformation as that given in [24]. The only difference is that the rules $p^+ \hookrightarrow_X p$ and $p^- \hookrightarrow_X \neg p$ are modalised with $X$ instead of BEL. However, this difference is flattened by the definition of social agents, where BEL converts universally and the all modalities are involved in conflicts.

*(2) Rule Loader* The rule loader builds a data structure as follows: for every atom $\alpha \in U^D$ we create three entries $\alpha$, INT$\alpha$ and OBL$\alpha$. Each entry has associated to it a list of hash tables:

For $\alpha$ we have

- $+h$ is a list of (pointers to) rules in $R^{\mathrm{BEL}}$ where $\alpha$ appears in the head;
- $-h$ is the list of rules in $R^{\mathrm{BEL}}$ where $\sim\alpha$ appears in the head;
- $+b$ is the list of rules in $R$ where $\alpha$ occurs in the body;
- $-b$ is the list of rules in $R$ where $\sim\alpha$ occurs in the body.

For $X\alpha$, $X \in \{\mathrm{INT}, \mathrm{OBL}\}$ we have

- $+h$ is a list of rules in $R^X$ where $\alpha$ appears in the head;
- $-h$ is the list of rules in $R^X$ where $\sim\alpha$ appears in the head;
- $+h^B$ is a list of rules in $R^{\mathrm{BEL}}$ where $\alpha$ appears in the head;
- $-h^B$ is a list of rules in $R^{\mathrm{BEL}}$ where $\sim\alpha$ appears in the head;
- $+b$ is the list of rules in $R$ where $X\alpha$ occurs in the body;
- $-b$ is the list of rules in $R$ where $X\sim\alpha$ occurs in the body.
- $+b_\sim$ is the list of rules in $R$ where $\sim X\alpha$ occurs in the body;
- $-b_\sim$ is the list of rules in $R$ where $\sim X\sim\alpha$ occurs in the body.

To each rule in $R^X$, $X \neq$ BEL, we associate a structure consisting of a (modal) literal (the head of the rule) and a set of pointers to the modal literals in the body of the rule, implemented as an hash table; while for belief rules we create the same structure as the other types of rules plus two other structures one for INT and one for OBL, the single pointer refers to the modal literal and the set of pointers corresponds to the literals in the body modalised, respectively, with INT and OBL.

*(3) The Inference Engine* The Inference Engine is based on an extension of the *Delores* algorithm/implementation proposed in [12] as a computational model of Basic Defeasible Logic. In turn

1. It asserts each fact (as an atom) as a conclusion and removes the atom from the rules where the atom occurs positively in the body, and it "deactivates" the rules where either the atom occurs negatively in the body, or incompatible modal literals occur in the body.
2. It scans the list of active rules for rules where the body is empty. It takes head and searches for rule (of the appropriate type) where the head is the negation of the atom or a modal literal incompatible with it. If there are no such rules then, the atom is appended to the list of facts, and removed from the rules.
3. It repeats the first step.
4. The algorithm terminates when one of the two steps fails. On termination the algorithm outputs the set of conclusions.[5]

It is immediate to see that the algorithm runs in linear time. Each (modal) atom/literal in a theory is processed exactly once and every time we have to scan the set of rules, thus the complexity of the above algorithm is $O(|U^D| * |R|)$.

Given the above result it might seem that social agents are computationally feasible. However, as we have seen in the previous sections there are situations (let us call them deviant situations) where social agents do not behave as expected. First of all, we have to identify when we have a deviant situation and what are the reasons why we have them, and what kind of control an agent has over them. Here we assume that a deviant situation depends on some primitive intentions of an agent (i.e., intentions given as facts). Since these intentions are independent of the policy the theory describe the only alternative a social agent has is to give up some of them. In the rest of the section we study whether this is possible and what price an agent has to pay to be social. The answer is negative; we will provide a theory that is essentially deviant, and we will show that social agents are (computationally) expensive.

A precise definition of the problem is provided in the next section.

[5] This algorithm outputs $\partial^+$; $\partial^-$ can be computed by an algorithm similar to this with the "dual actions". For $\Delta^+$ we have just to consider similar constructions where we examine only the first parts of step 1 and 2. $\Delta^-$ follows from $\Delta^+$ by taking the dual actions.

### 5.3 Restoring Sociality Problem

INSTANCE:
Let $I$ be a finite set of primitive intentions, OBL$p$ a primary obligation, and $D$ a theory such that $I \subseteq F$, $D \vdash -\partial_{\text{OBL}}p$, $D \vdash -\Sigma_{\text{OBL}}{\sim}p$, $D \vdash +\partial_{\text{INT}}{\sim}p$, $D \vdash +\Sigma_{\text{OBL}}p$ and $D \vdash -\Sigma_{\text{BEL}}{\sim}p$.
QUESTION:
Is there a theory $D'$ equal to $D$ apart from containing only a proper subset $I'$ of $I$ instead of $I$, such that $\forall q$ if $D \vdash +\partial_{\text{OBL}}q$ then $D' \vdash \partial_{\text{OBL}}q$ and $D' \vdash +\partial_{\text{OBL}}p$?

The specification of the problem is meant to formalise the situation we have described in the previous sections. The combination of the proof tags in the specification of the instance is only possible in case there is an applicable deontic rule for $p$ ($+\Sigma_{\text{OBL}}p$) which would be otherwise unchallenged, i.e., there are no deontic rules to support ${\sim}p$ ($-\Sigma_{\text{OBL}}{\sim}p$) and there are no reasons to believe the opposite, is defeated, against the sociality of the agent, by the intentionality of ${\sim}p$ obtained as a consequence of an intention of the agent (this means it has been obtained by converting a belief rule into an intention rule). In other terms a potentially valid obligation is blocked by a consequence of an intentional behaviour.

**Example 5** Let us the consider the theory consisting of

$$
\begin{aligned}
F &= \{\text{INT}p, \text{INT}s\} \\
R &= \{r_1 : p, s \Rightarrow_{\text{BEL}} q \quad r_2 : \Rightarrow_{\text{OBL}} {\sim}q \quad r_3 : \Rightarrow_{\text{BEL}} s\} \\
> &= \{r_1 > r_2\}
\end{aligned}
$$

$r_1$ is a belief rule and so the rule is stronger than the deontic rule $r_2$. In addition we have that the belief rule is not applicable (i.e., $-\Sigma_{\text{BEL}}q$) since there is no way to prove $+\partial_{\text{BEL}}p$. There are no deontic rules for $q$, so $-\partial_{\text{OBL}}q$. However, rule $r_1$ behaves as an intention rule since all its antecedent can be proved as intentions, i.e., $+\partial_{\text{INT}}p$ and $+\partial_{\text{INT}}s$. Hence, since $r_1$ is stronger than $r_2$, the derivation of $+\partial_{\text{OBL}}{\sim}q$ is prevented against the sociality of the agent.

The related decision problem is whether it is possible to avoid the "deviant" behaviour by giving up some primitive intentions, retaining all the (primary) obligations, and maintaining a set of primitive intentions as close as possible to the original set of intentions.

**Example 5** (CONTINUED). When we examine the theory we notice that both primitive intentions concur to the prevention of the derivation of $+\partial_{\text{OBL}}{\sim}q$. These intentions are under the control of the agent. The agent has the opportunity to avoid the deviant behaviour if she gives up at least one of her primitive intentions. Accordingly, the agent has three alternatives: to give up INT$p$, to give up INT$s$, or to give up both. The first two options minimise the difference between the original theory and the resulting theory.

There could be cases where, no matter what intentions are removed, the theory will result in a deviant situation. The simplest case is where there are intentions that are at the same time primitive and primary.

**Example 6** Let the theory $D$ be

$$\begin{aligned} F &= \{\mathrm{INT}p\} \\ R &= \{r_1 : \Rightarrow_{\mathrm{INT}} p \quad r_2 : p \Rightarrow_{\mathrm{BEL}} q \quad r_3 : \Rightarrow_{\mathrm{OBL}} {\sim}q\} \\ > &= \{r_2 > r_3\} \end{aligned}$$

In this theory we have only one primitive intention and therefore the only way to see whether it is possible to avoid the problem is to give up that intention. However, we have that $r_1$ is an intention rule for $p$, and thus we can use it to derive $+\partial_{\mathrm{INT}}p$, which allows $r_2$ to be used to derive an intention instead of a belief, and consequently to prevent the derivation of an obligation against the sociality of the agent.

Notice that, given the non-monotonic nature of defeasible logic, it is possible that a solution to the problem is given by a superset of the original set of intentions instead of a subset.

**Example 7** Given a theory $D$ as follows

$$\begin{aligned} F &= \{\mathrm{INT}a, \mathrm{INT}b\} \\ R^{\mathrm{BEL}} &= \{r_1 : \mathrm{INT}a \Rightarrow_{\mathrm{BEL}} d, \quad r_2 : \mathrm{INT}b \Rightarrow_{\mathrm{BEL}} d, \\ &\qquad r_3 : \mathrm{INT}c \Rightarrow_{\mathrm{BEL}} {\sim}d, \quad r_4 : d \Rightarrow_{\mathrm{BEL}} e\} \\ R^{\mathrm{INT}} &= \{r_5 : \Rightarrow_{\mathrm{INT}} a, \quad r_6 : \Rightarrow_{\mathrm{INT}} b\} \\ R^{\mathrm{OBL}} &= \{r_7 : \Rightarrow_{\mathrm{OBL}} {\sim}e\} \\ > &= \{r_3 > r_1, r_3 > r_2, r_4 > r_7\} \end{aligned}$$

As we have seen in the previous example, throwing away the two primitive intentions is of no avail, they are reinstated by the intention rules $r_5$ and $r_6$. However, to block the side effect $d$ of the two intentions we can introduce a further primitive intention, INT$c$.

If we replace the theory $D$ by a theory $D'$ obtained from $D$ by emptying the set of intention rules, then we have two alternatives to avoid the deviance. The first is to drop both the primitive intentions INT$a$ and INT$b$, or we can form a new primitive intention INT$c$. In this case the theory obtained from adding the new intention is, intuitively, more similar to the original theory than the theory obtained from dropping the two primitive intentions.

Variations of the problem can be obtained by changing other parameters of the specification. Some of these can define new types of agents. For example a *pro-active social agent* might try to recover from a deviant situation by changing the raw facts (facts that are neither primitive intentions nor primitive obligations). Thus a pro-active social agent tries to adapt the environment to her goals

(intentions). A legalistic social agent, on other the hand, might change the set of primitive obligations, while a cheating social agent might change the rules. However, it is important to realise that all these variations have a structure isomorphic to the specification we discuss in this paper. In addition it is possible to generalise the problem to the case of multiple deviant behaviours.

**Theorem 4.** *The Restoring Sociality Problem is NP-complete.*

*Proof.* We have to show that the problem is both NP and NP-hard. For the NP part all we have to do is to notice that we can guess a theory, we compute the extension of the theory in linear time (Theorem 1) and then verify in linear time whether the restore conditions are satisfied.

For the NP-hard part we have to map a known NP-complete problem to the Restoring Sociality Problem. Here we use the *knapsack problem* [29, Problem MP9].

*Knapsack Problem*

> INSTANCE:
> Given a finite set $U$, for each $u \in U$ a size $s(u) \in \mathbb{Z}^+$ and a value $v(u) \in \mathbb{Z}^+$, and integer $B$ and $K$.
> QUESTION:
> Is there a subset $U' \subseteq U$ such that $\sum_{u \in U'} s(u) \leq B$ and $\sum_{u \in U'} v(u) \geq K$?

The knapsack problem is encoded by a defeasible theory $D$ where $R$ is as follows:

- $\mathrm{INT}load(u) \Rightarrow_{\mathrm{BEL}} load(u)$ for each $u \in U$.
- $\sum_{s(u):D \vdash +\partial_{\mathrm{BEL}} load(u)} s(u) > B \Rightarrow_{\mathrm{INT}} overload$
- $\sum_{s(u):D \vdash +\partial_{\mathrm{BEL}} load(u)} v(u) < K \Rightarrow_{\mathrm{INT}} undervalue$
- $overload \Rightarrow_{\mathrm{BEL}} \neg good$
- $undervalue \Rightarrow_{\mathrm{BEL}} \neg good$
- $\Rightarrow_{\mathrm{OBL}} good$

$F$ is given by the relationship $\mathrm{INT}load(u) \in F$ iff $u \in U'$.

The theory of the above construction has several interesting properties. First of all $D \vdash +\partial_{\mathrm{BEL}} load(u)$ iff $\mathrm{INT}load(u) \in F$, which means $u \in U'$; then $D \vdash +\partial_{\mathrm{OBL}} good$ iff either of the two conditions of the knapsack problem are satisfied; notice that since there are no literals for $\neg load(u)$, the computation of the rule $\mathrm{INT}load(u) \Rightarrow_{\mathrm{BEL}} load(u)$ can be computed independently of the rest of the theory thanks to the modularity of DL [24], thus the sums in the antecedent of the second and third rule can be considered as "facts" in the theory. In case one of the condition of the knapsack problem is not satisfied we have exactly a deviant situation as in the restoring sociality problem. The encoding of the knapsack problem in DL is clearly linear, thus any algorithm that solves the restoring sociality problem in polynomial time will solve the knapsack problem in polynomial time. Therefore the restoring sociality problem is NP-complete.

### 5.4 Revising Deviant Situations

In this paper we focused on what we called social agents, i.e., agents who refrain from planning activities which may result in a violation of existing obligations. However, we would like to stress out that the so called "restoring sociality problem", and the computational complexity results associated with it, is not specific to social agents, but it depends on the structure of an agent type. In particular any agent type defined by the following parameters

$$\text{Convert}(X, Y), \text{Conflict}(X, Z), \text{Conflict}(X, Y), \text{Conflict}(Z, Y)$$

suffers from the same problem (of course with a different intuitive reading of the problem).

In a similar way the transformations to remove defeaters and to empty the superiority relation, as well as the general complexity result for the logic obtain for all agent types (modal defeasible logic variants) isomorphic to social agents.

A first solution to the complexity of social agents is to avoid conversions. However, we believe that this is a rather unsatisfactory approach for agents with both internal (intentions) and external (obligations) motivational attitudes. It is not possible to capture the notion of intentionality which is of paramount importance when we deal with agents situated in normative contexts.

A second solution would be to assume that belief rules behaving as intention rules (i.e., obtained from the conversion $\text{Convert}(\text{BEL}, \text{INT})$) are always weaker than deontic rules or belief rules behaving as deontic rules (i.e., where the conversion $\text{Convert}(\text{BEL}, \text{OBL})$ applies). In this case the problem is with theory like

$$\begin{array}{ll} r_1 : a \Rightarrow_{\text{BEL}} q & r_2 : b \Rightarrow_{\text{BEL}} \sim q \\ +\partial_{\text{INT}} a & +\partial_{\text{OBL}} b \\ r_1 > r_2 & \end{array}$$

where $r_1$ is at the same time stronger and weaker than $r_2$.

## 6 Related Work

As was mentioned in the introduction, reasoning about mental attitudes is a central issue in philosophy and AI. Despite the plethora of proposals devoted to this topic, the related work that is directly relevant for this paper is mainly the BOID architecture. The basic calculation scheme used in BOID [4] is similar to the one proposed in this paper. It is worth noting that in [16] we show that BOID is a particular case of the logic of [5] (and of a particular version of the logic of [7,6] without the $\otimes$ operator). This is due to the use adopted there of the superiority relation ($>$). However, the framework of the current paper is closer to BOID, as we distinguish conflicts between rules for the same modality and for different modalities. In the second case, the relation $\text{Conflict}(X, Y)$ assumes that $X$ rules are always stronger than $Y$'s.

The BOID framework has four components representing respectively the beliefs (B), obligations (O), intentions (I) and desires (D) of the agent. The behaviour of each component is specified by sets of propositional logical formulas

often in the form of defeasible rules. BOID identifies two general types of conflicts that could arise either within each component (*internal conflicts*) or between the components (*external conflicts*). These two types of general conflicts are further subdivided into different subtypes which gives rise to several possible conflicts among the mental attitudes. In order to solve possible conflicts among the attitudes an ordering function ($\rho$) is defined on rules based on the *agent type*. An agent type is determined by allowing one component to overrule others. For example, a *realistic* agent type can be defined by having an ordering in which the belief component overrules any other component (BOID, BODI, BDIO etc.). This means that in BOID a conflict resolution type is an order of overruling and in general the order of derivation can be used to identify different types of agents. Agent types like *simple-minded* (agent type where prior intentions overrule desires and obligations), *social* (agent type where obligations overrule desires) etc. could be defined in a similar manner. Formally an agent type is defined as a function, $\rho$ that assigns a unique integer to each rule. It should be noted that the ordering function $\rho$ assigns unique values to the rules of all components such that the values of all rules from one component are either smaller or greater than the values of all rules from another component.

Besides the specific result discussed in Section 5.3, the general aspects that differentiate the current framework from BOID's are the following:

- our proof conditions permit to derive modalised literals; accordingly, in addition to labelling rules by the elements of MOD, modalities are also made explicit in rule antecedents, thus enriching the expressive power of the logic;
- conversions are introduced to capture some fundamental reasoning patterns which, in most cases, should be admitted or which may in any case contribute to characterise agent types;
- we admit that Conflict may cover only some modalities; this makes it possible that, for any rule types $X$ and $Y$ that are not covered by Conflict, we can obtain $+\partial_X p$ and $+\partial_Y \sim p$;
- our logic for BIO agents has linear complexity, whereas to our knowledge there is no analogous result for BOID.

## 7 Summary

The contribution of this paper is manifold. We extend the analysis of policy-based cognitive agents with the notion of obligation and we argue that in such case side-effects do not endanger the logical analysis but on the contrary are beneficial to explain notions, e.g., intentionality, of paramount importance for agents situated in normative and legal contexts.

Policy based agents are represented in DL by extendeding the logic with the modalities of belief, intention and obligation. This choice was motivated by the computational feasibility of the logic. We have demonstrated that the logic has linear complexity. As far as we know this is the first result of this kind for cognitive agents.

Finally we have studied the notion of social agent and we have proved that a proper and philosophically sound treatment of this notion leads to an increase of the computational complexity of the problem. Again this is the first result of this kind we are aware of. As we argued, the result can be easily extended to all the other agent types for which we can have deviant situations.

## Acknowledgements

The first author was supported by the Australian Research Council under the Discovery Project DP0558854.

## References

1. Bratman, M., Israel, D., Pollack, M.: Plans and resource-bounded practical reasoning. Computational Intelligence **4** (1988) 349–355
2. Cohen, P., Levesque, H.: Intention is choice with commitment. Artificial Intelligence **42** (1990) 213–261
3. Rao, A.S., Georgeff, M.P.: Modelling rational agents within a BDI-architecture. In: Proceedings of the Second International Conference on Principles of Knowledge Representation and Reasoning (KR'91), Morgan Kaufmann (1991) 473–484
4. Broersen, J., Dastani, M., Hulstijn, J., van der Torre, L.: Goal generation in the BOID architecture. Cognitive Science Quarterly **2** (2002) 428–447
5. Governatori, G., Rotolo, A.: Defeasible logic: Agency, intention and obligation. In Lomuscio, A., Nute, D., eds.: Deontic Logic in Computer Science. Number 3065 in LNAI, Berlin, Springer-Verlag (2004) 114–128
6. Dastani, M., Governatori, G., Rotolo, A., van der Torre, L.: Programming cognitive agents in defeasible logic. In Sutcliffe, G., Voronkov, A., eds.: Proc. LPAR 2005. Volume 3835 of LNAI., Springer (2005) 621–636
7. Dastani, M., Governatori, G., Rotolo, A., van der Torre, L.: Preferences of agents in defeasible logic. In Zhang, S., Jarvis, R., eds.: Proc. Australian AI05. Volume 3809 of LNAI., Springer (2005) 695–704
8. Dignum, F.: Autonomous agents with norms. Artificial Intelligence and Law **7** (1999) 69–79
9. Dignum, F., Morley, D., Sonenberg, L., Cavedon, L.: Towards socially sophisticated BDI agents. In: ICMAS (4th International Conference on Multi-Agent Systems). (2000) 111–118
10. Pitt, J., ed.: Open Agent Societies. Wiley, Chichester (2005)
11. Thomason, R.H.: Desires and defaults: A framework for planning with inferred goals. In Cohn, A.G., Giunchiglia, F., Selman, B., eds.: KR2000, San Francisco, Morgan Kaufmann (2000)
12. Maher, M.J., Rock, A., Antoniou, G., Billignton, D., Miller, T.: Efficient defeasible reasoning systems. International Journal of Artificial Intelligence Tools **10** (2001) 483–501
13. Bassiliades, N., Antoniou, G., Vlahavas, I.: DR-DEVICE: A defeasible logic system for the Semantic Web. In Ohlbach, H.J., Schaffert, S., eds.: 2nd Workshop on Principles and Practice of Semantic Web Reasoning. Number 3208 in LNCS, Springer (2004) 134–148

14. Antoniou, G., Billington, D., Governatori, G., Maher, M.J.: A flexible framework for defeasible logics. In: Proc. American National Conference on Artificial Intelligence (AAAI-2000), Menlo Park, CA, AAAI/MIT Press (2000) 401–405
15. Bratman, M.E.: Intentions, Plans and Practical Reason. Harvard University Press, Cambridge, MA (1987)
16. Governatori, G., Padmanabhan, V., Rotolo, A., Sattar, A.: A defeasible logic for modelling policy-based intentions and motivational attitudes. Journal of Applied Logic (2007), submitted.
17. Nute, D., ed.: Defeasible Deontic Logic. Kluwer, Dordrecht (1997)
18. Sartor, G.: Legal Reasoning: A Cognitive Approach to the Law. Springer, Dordrecht (2005)
19. Girle, R.: Modal Logic and Philosophy. Acumen, Teddington (2000)
20. von Wright, G.H.: Norm and Action. Routledge, London (1963)
21. Nute, D.: Defeasible reasoning. In: Proceedings of 20th Hawaii International Conference on System Science, IEEE press (1987)
22. Nute, D.: Defeasible logic. In: Handbook of Logic in Artificial Intelligence and Logic Programming. Volume 3. Oxford University Press (1987)
23. Billington, D.: Defeasible logic is stable. Journal of Logic and Computation **3** (1993)
24. Antoniou, G., Billington, D., Governatori, G., Maher, M.J.: Representation results for defeasible logic. ACM Transactions on Computational Logic **2** (2001) 255–287
25. Maher, M.J., Rock, A., Antoniou, G., Billignton, D., Miller, T.: Efficient defeasible reasoning systems. International Journal of Artificial Intelligence Tools **10** (2001)
26. Maher, M.J.: Propositional defeasible logic has linear complexity. Theory and Practice of Logic Programming **1** (2001) 691–711
27. Nute, D.: Norms, priorities, and defeasible logic. In McNamara, P., Prakken, H., eds.: Norms, Logics and Information Systems. IOS Press, Amsterdam (1998) 201–218
28. Kraus, S., Lehmann, D., Magidor, M.: Nonmonotonic reasoning, preferential models and cumulative logics. Artificial Intelligence **44** (1990) 167–207
29. Garey, M., Johnson, D. In: Computers and Intractability: A Guide to the Theory of NP-Completeness. W. H. Freeman and Company (1979)

# Choosing Your Beliefs

Guido Boella[1], Célia da Costa Pereira[2], Gabriella Pigozzi[3], Andrea Tettamanzi[2], and Leendert van der Torre[3]

[1] Università di Torino, Dipartimento di Informatica
10149, Torino, Cso Svizzera 185, Italia
guido@di.unito.it
[2] Università degli Studi di Milano, Dipartimento di Tecnologie dell'Informazione
26013, Crema, via Bramante 65, Italy
pereira@dti.unimi.it, andrea.tettamanzi@unimi.it
[3] Université du Luxembourg, Computer Science and Communication
L-1359, Luxembourg, rue Richard Coudenhove - Kalergi 6, Luxembourg
gabriella.pigozzi@uni.lu, leendert@vandertorre.com

**Abstract** This paper presents and discusses a novel approach to indeterministic belief revision. An indeterministic belief revision operator assumes that, when an agent is confronted with a new piece of information, it can revise its belief sets in more than one way. We define a rational agent not only in terms of what it believes but also of what it desires and wants to achieve. Hence, we propose that the agent's goals play a role in the choice of (possibly) one of the several available revision options. Properties of the new belief revision mechanism are also investigated.

**Keywords.** Rational agents, indeterministic belief revision, qualitative decision theory.

## 1 Motivating example

Suppose that you believe that:

1. A liberal policy leads to decrease of unemployment, and
2. A decrease of unemployment leads to re-election,

and you desire to be re-elected. Therefore you execute a plan based on a liberal policy, or do something else to decrease unemployment, and secure your re-election.

Now suppose that someone informs you that a liberal policy does not lead to re-election. Assume the person telling you this is very trustworthy, has a good reputation, so you believe what he is telling you. This implies that the three beliefs cannot hold together, and you have to give up one of them.

Now assume in addition that when you give up your belief in the first rule, you still have another plan to decrease unemployment, and thus to be re-elected, for example by increasing government spending on public works like building

Dagstuhl Seminar Proceedings 07122
Normative Multi-agent Systems
http://drops.dagstuhl.de/opus/volltexte/2007/938

bridges. However, if you give up your second belief that lower unemployment leads to re-election, you do not have an alternative plan to achieve re-election. In that case, you might reason by cases as follows.

1. Let us first assume that the first belief that liberal policy leads to lower unemployment is factually wrong, whereas the second belief is true. If you choose to retain the first (wrong) belief and to reject the second one, then you will do nothing and you will not succeed in being re-elected. But, had you kept your belief in the second rule and rejected the first belief, you could have increased public spending in order to decrease unemployment, and therefore you could have achieved your goal to be re-elected. To conclude, choosing to maintain the first belief, you risk to miss a goal you could have achieved.
2. Let us now assume assume that the first belief is actually true and the second belief is wrong. If you choose to keep the second (wrong) belief that decreasing unemployment leads to re-election, you will increase public spending, but you will not achieve the goal of being re-elected. However, even if you had chosen the right revision, i.e. to retain the first belief and reject the second one, you could have not achieved your goal of re-election. To conclude, by choosing the second (wrong) rule, you believed you could achieve a goal when you could not, so you will be disappointed for trying in vain but at least you tried to reach your aim.

The moral of the story is that, if you are interested only in realizing your goal (and there are no other goals relevant for you), then choosing the second belief — *even when it is factually wrong* — is the only rational choice. This is because, independently of the second belief being right or wrong, by choosing that belief you will end up in an optimal state. Moreover, in one situation — the first one — you will end up in a better state if you choose the second belief than if you choose the first one. Summarizing, you should drop the first belief, because in that way, you keep open all possibilities to achieve your goal.

We can formalize the above example, by defining the following atomic propositions:

$p$ you are following a liberal policy;
$u$ unemployment decreases;
$r$ you will be re-elected;
$s$ you are increasing public spending.

The belief base before being told that a liberal policy does not lead to re-election ($\neg(p \supset r)$) would contain the three formulas $p \supset u$, $u \supset r$, and $s \supset u$. You desire, first of all, to be re-elected, $r$, and, if possible, not to increase public spending, $\neg s$. Adding $\neg(p \supset r)$ to your beliefs would make them inconsistent. Therefore, you have to revise your beliefs by giving up either $p \supset u$ or $u \supset r$. The choice you make may depend on the goals you can achieve in the alternatives: if you give up $p \supset u$, your plan will be to increase public spending, so you will not achieve $\neg s$, but might succeed in achieving $r$; if you give up $u \supset r$, your plan will be to do nothing, so you will certainly not achieve $r$, but you will fulfill $\neg s$.

Depending on the payoff you expect from $r$ and $\neg s$, you could prefer one or the other alternative.

We use the re-election example as a running example throughout the paper. There are some particular issues involved in the re-election situation, such as temporal references. However, we believe that many problems can be phrased in a similar way, including examples referring to factual statements about the present state of the world rather than hypothetical statements referring to the future as in the re-election example. For example, $p$ may stand for "it is a public holiday", $q$ for "your favorite restaurant is open", $r$ stands for "eating in the restaurant", and you are informed that the restaurant is closed on public holidays. In that case, you would give up your belief that it is a public holiday, because that is the only way to achieve your goal to eat in the restaurant. You may drive to the restaurant in vain, but it would be much worse to give up the belief that the restaurant is open, only to find out later that you could have eaten there. At least, as long as we assume that this line of reasoning does not interfere with other goals. For example, the goal not to drive to restaurants in vain should not be preferred to the goal to eat in the restaurant, and there should not be information that it is much more likely that it is a public holiday than that the restaurant is open, and so on.

The choice among belief sets is distinct from other decision problems, due to the possibility of wishful thinking. Consider for example that you desire that liberal policy leads to lower unemployment, and that this desire is preferred to the desire to be re-elected. What will you do? At least in a naive approach, you could reason by cases as follows. Assume that you choose the first belief, in that case you believe that will achieve the desire that liberal policy leads to lower unemployment. Assume that you choose the second belief, in that case you believe that you will achieve the goal to be re-elected. Since the first goal is more important than the second one, you choose the first belief. Analogously, in the restaurant example, if you like public holidays and this desire is stronger than your desire to eat in the restaurant, than a naive reasoner may choose the first belief set. However, this is again a case of wishful thinking, and not a valid reason not to go to the restaurant. Instead, one should reason by cases as follows. Either it is a public holiday or not. If it is a public holiday, it does not matter what we do, we always achieve the goal that it is a public holiday and we never achieve the goal to go to the restaurant, since it is closed. If it is not a public holiday, then we will never achieve the desire for public holidays, but we may achieve the desire to eat in the restaurant — at least, when we go there. Our formal framework illustrates why the line of reasoning leading to the decision not to go to the restaurant is fallacious.

The idea of this paper is inspired by the notion of *conventional wisdom* (CW) as introduced by economist John Kenneth Galbraith:

> We associate truth with convenience, with what most closely accords with self-interest and personal well-being. ([14], p. 34)

That is, CW consists of "ideas that are convenient, appealing". This is the rationale for keeping them. One basic brick of CW could then be the fact that

some ideas are maintained because they maximize the goals that the agents (believe) they can achieve. This work may be seen as an initial attempt to formally capture the concept of a CW agent. In the following we provide a logical framework that models how a CW agent revises its beliefs.

The paper is structured as follows. In Section 2 we introduce the aim of this paper, the used methodology and particular challenges encountered. In Section 3 we introduce the agent theory we use in our approach, and in Section 4 we introduce an indeterministic belief change operator in this agent theory. In Section 5 we define the choice among beliefs as a decision problem in the agent theory. We conclude the paper by clarifying the relation of our proposal with some existing work (Section 6) and with some final remarks and ideas for future research (Section 7).

## 2 Aim, methodology and challenges

The research problem of this paper is to develop a formal model to reason about the kind of choices among belief sets discussed in the previous section, and to generalize the example above in case of additional beliefs, multiple goals with preferences among them, conditional desires, a way to take violated goals into account, and so on.

We use a combination of the framework of belief revision together with a qualitative decision theory. Classical approaches to belief revision assume that, when an agent revises its belief set in view of a new input, the outcome is well-determined. This picture, however, is not realistic. When an agent revises its beliefs in the light of some new fact, it often has more than one available alternative. Approaches to belief revision that do not stipulate the existence of a single revision option are called *indeterministic* [16,19]. In this paper we suggest that one possible policy an agent can use in order to choose among available alternatives is to check the effect of the different revisions on the agent's set of goals.

Moreover, for the qualitative decision theory we are inspired by agent theories such as the BOID architecture [2], the framework of goal generation in 3APL as developed by van Riemsdijk and colleagues [20], and [5]. In particular, our agent model is based on one of the versions of 3APL, because the belief base in the mental state of a 3APL agent is a consistent set of propositional sentences, just like in the framework of belief revision. However, we do not include "planning rules" or "practical reasoning rules" representing which action to choose in a particular state, because we aim for a modular agent architecture. We assume that there is a planning module, which would take a set of goals, actions, and an initial world state representation in input and produce a solution plan in output. This planning module might rely on the well-known graphplan algorithm, or any other AI planner: as in object-oriented programming, we *encapsulate* the planner within a well-defined interface and overlook the implementation details of how a solution plan is found. This is in line with the BOID architecture [2], where the planning component is kept separate from the remainder of agent deliberation.

In other words, we model the choice among belief sets essentially as a decision problem, that is, as a choice among a set of alternatives. We do not use classical decision theory (utility function, probability distribution, and the decision rule to maximize expected utility), but a qualitative version based on maximizing achieved goals and minimizing violated goals in an abstract agent theory (see e.g. [8] for various approaches to formalize the decision process of what an agent should do), because such qualitative decision theories include beliefs and therefore are easier to combine with the theory of belief revision. However, what precisely are the alternatives?

An indeterministic belief revision operator associates multiple revision options to a belief set that turns out to be inconsistent as a consequence of a new piece of information. Our revision mechanism selects the revision alternative that allows the agent to maximize its achievable goals. However, it will not always be possible to select exactly one revision alternative. For example, there may be one preferred goal set but two revision alternatives that lead the agent to achieve it. In this case, the two belief revision candidates are said to be equivalent. In Section 5.3 we will provide conditions under which a revision for a CW agent is deterministic, that is, when our revision operator can select exactly one revision alternative.

Besides the issue of wishful thinking, another complicating factor when choosing among belief sets in the context of conditional desire rules, is that a maximization of goals may lead to a meta-goal to derive goals. However, deriving goals by itself does not have to be desirable. In contrast, it may even be argued that fewer goals is better than more goals, as you risk to violate goals and become unhappy (as in Buddhism). One possible solution would be taking also goal violations into account. However, we do not address this issue in this paper.

## 3 An abstract agent theory

In this section, we represent the formalism which is used throughout the paper.

### 3.1 A brief introduction to AI planning and agent theory

Any agent, be it biological or artificial, must possess knowledge of the environment it operates in, in the form of e.g. *beliefs*. Furthermore, a necessary condition for an entity to be an *agent* is that it acts. We shall call the factors that motivate an agent to act *desires*. For artificial agents, desires may be the purposes an agent was created for.

Desires are necessary, not sufficient, conditions for action. When a desire is met by other conditions that make it possible for an agent to act, that desire becomes a *goal*.

The reasoning side of acting is known as practical reasoning or deliberation, which may include *planning*. Planning is a process that chooses and organizes actions by anticipating their expected effects with the purpose of achieving as good as possible some pre-stated objectives or goals.

Acting does not always imply planning. An agent deliberates or "chooses" to plan when it has to make difficult or unusual tasks; or when there are high risks or interests; or when there is the necessity of synchronizing several tasks which are part of a dynamic system. Otherwise, its acting may be based on simple stimulus-response rules.

Formally, AI planning may be defined as follows. Disposing of

(i) a representation of the initial state of the world $I$;
(ii) a set $A$ of actions; and
(iii) a description of the goals to be reached $G$;

planning consists of finding a sequence of actions from $A$ — *plan* — whose execution, from the initial state $I$, leads to a final state in which $G$ is satisfied. Each action has a set of conditions it needs to be executed — *preconditions*— and once executed, it produces one or more *effects* which changes the world from a state to another. One of the theoretical motivations for planning is its utilization as a component of the rational behavior of an agent. In this case, this is all about providing the agent which has the task of constructing and/or executing a plan, reasoning capabilities like being able to react according to its perceptions, i.e., according to the changes in its mental state (beliefs and desires). In this case, planning consists in constructing a plan whose execution leads from the initial state to a final state which satisfies the *goals emerged during the planning process or during the plan execution.*

In the example, we may assume you dispose of two actions:

$a_p$ (implement a liberal policy), with no precondition, and with effect $p$;
$a_s$ (increase public spending), with no precondition, and with effect $s$.

Our formalism is inspired by one of the variants of the agent programming language 3APL used in [20]. However, unlike [20], the objective of our formalism is to analyze, not to develop, agent systems. More precisely, our agent must single out the *best* set of goals to be given as an input to a traditional planner. That is because the intentions of the agent are not considered. We merely consider beliefs (knowledge the agent has about the world states), desires (or motivations) and relations (desire-adopting rules) defining how the desire base will change with the acquisition of new beliefs and/or new desires. The goal generation process that underlies this work is very much in line with the work carried out in [21] on *oversubscription planning problems*, in which the main objective is to find the maximal set of desires to be reached in a given period and with a limited quantity of resources, and with goal generation in the BOID architecture [2].

### 3.2 Beliefs, Desires, and Goals

The basic components of our language are *beliefs* and *desires*. Beliefs are represented by means of a *belief base*. A belief base is a finite and consistent set of propositional formulas describing the information the agent has about the world and internal information. Desires are represented by means of a *desire base*. A

desire base consists of a set of propositional formulas which represent the situations the agent would like to achieve. However, unlike the belief base, a desire base may be inconsistent, e.g., $\{p, \neg p\}$.

**Definition 1 (Belief Base $\Sigma$ and Desire Base $\Gamma$)** *Let $\mathcal{L}$ be a propositional language with $\top$ a tautology, and the logical connectives $\wedge$ and $\neg$ with the usual meaning. The agent's belief base $\Sigma$ is a consistent finite set of atomic propositions like $\phi$, $\varphi$, $\psi$, ... and compound propositions like $\neg\phi$, $\varphi \wedge \psi$, and so on. $\Sigma$ can also be represented as the conjunction of its propositional formulas. The agent's desire base is a possibly inconsistent finite set of sentences denoted by $\Gamma$, with $\Gamma \subseteq \mathcal{L}$.*

We use modal languages to talk about the belief and desire bases of the agent. Since the belief and desire bases of an agent are completely separated, there is no need to nest the operators **B** and **D**.

**Definition 2 (Belief Formulas $\beta$ and Desire Formulas $\kappa$)** *Given any formula $\phi$ of $\mathcal{L}$, $\mathbf{B}\phi$ means that $\phi$ is believed whereas $\mathbf{D}\phi$ means that $\phi$ is desired. The languages of belief formulas $\beta \in \mathcal{L}_B$ and desire formulas $\kappa \in \mathcal{L}_D$ are defined as follows:*

$$\beta ::= \top | \mathbf{B}\phi | \neg\beta | \beta_1 \wedge \beta_2$$

$$\kappa ::= \top | \mathbf{D}\phi | \neg\kappa | \kappa_1 \wedge \kappa_2$$

Following van Riemsdijk and colleagues, the antecedent of a desire-adoption rule consists of a belief condition and a desire condition; the consequent is a propositional formula. Intuitively, this means that if the belief and the desire conditions in the antecedent hold, the formula in the consequent is automatically adopted as a desire. Note that this implies that in the antecedent we may have for example a disjunction of two beliefs or a disjunction of two desires, but we cannot have a disjunction of a belief and a desire.

**Definition 3 (Desire-Adoption Rules $\mathcal{R}_D$)** *The language of desire-adoption rules $\mathcal{L}_R$ is defined as follows:*

$$\mathcal{L}_R = \{\beta, \kappa \Rightarrow_D^+ \phi \mid \beta \in \mathcal{L}_B, \kappa \in \mathcal{L}_D, \phi \in \mathcal{L}\}$$

*The set of desire-adoption rules $\mathcal{R}_D$ of an agent is a finite subset of $\mathcal{L}_R$.*

Goals, in contrast to desires, are represented by consistent desire bases. There are various ways to generate candidate goal sets from the desire adoption rules, as discussed in the remainder of this section.

**Definition 4 (Candidate Goal Set $\Gamma^*$)** *A candidate goal set $\Gamma^*$ is a consistent subset of $\Gamma$.*

### 3.3 Mental State Representation

We assume that an agent is equipped with three bases:

- belief base $\Sigma \subseteq \mathcal{L}$;
- desire base: $\Gamma \subseteq \mathcal{L}$;
- desire-adoption rule base $\mathcal{R}_D$;

The state of an agent is completely described by a triple $\mathcal{S} = \langle \Sigma, \Gamma, \mathcal{R}_D \rangle$. In addition, we assume that each agent can be described using a $\mathcal{P}$-dependent function $\mathcal{F}_\mathcal{P}$, a pay-off function $f : \mathcal{L} \to \mathbb{R}$, a goal selection function $G$, and a belief revision operator $*$, as discussed below.

In the example,

$$\begin{aligned} \Sigma &= \{\neg(p \wedge \neg u), \neg(u \wedge \neg r), \neg(s \wedge \neg u)\}, \\ \Gamma &= \{r, \neg s\}, \\ \mathcal{R}_D &= \{\top, \top \Rightarrow^+_D r; \top, \top \Rightarrow^+_D \neg s\}. \end{aligned}$$

The semantics we adopt for the belief formulas is standard.

**Definition 5 (Semantics of Belief Formulas)** *Let $\phi \in \mathcal{L}$, $\beta \in \mathcal{L}_B$, and let $\langle \Sigma, \Gamma, \mathcal{R}_D \rangle$ be the mental state of an agent. The semantics of belief formulas is given as*

$$\begin{aligned} &\langle \Sigma, \Gamma, \mathcal{R}_D \rangle \models_{\mathcal{L}_B} \top \\ &\langle \Sigma, \Gamma, \mathcal{R}_D \rangle \models_{\mathcal{L}_B} \mathbf{B}\phi \Leftrightarrow \Sigma \models \phi \\ &\langle \Sigma, \Gamma, \mathcal{R}_D \rangle \models_{\mathcal{L}_B} \neg\beta \Leftrightarrow \langle \Sigma, \Gamma, \mathcal{R}_D \rangle \not\models_{\mathcal{L}_B} \beta \\ &\langle \Sigma, \Gamma, \mathcal{R}_D \rangle \models_{\mathcal{L}_B} \beta_1 \wedge \beta_2 \Leftrightarrow \langle \Sigma, \Gamma, \mathcal{R}_D \rangle \models_{\mathcal{L}_B} \beta_1 \text{ and } \langle \Sigma, \Gamma, \mathcal{R}_D \rangle \models_{\mathcal{L}_B} \beta_2 \end{aligned}$$

The semantics we adopt for desire formulas is similar to the semantics of goal formulas proposed in [20].

**Definition 6 (Semantics of Desire Formulas)** *Let $\phi \in \mathcal{L}$, $\kappa \in \mathcal{L}_D$, and let $\langle \Sigma, \Gamma, \mathcal{R}_D \rangle$ be the mental state of an agent. The semantics of desire formulas is given as*

$$\begin{aligned} &\langle \Sigma, \Gamma, \mathcal{R}_D \rangle \models_{\mathcal{L}_D} \top \\ &\langle \Sigma, \Gamma, \mathcal{R}_D \rangle \models_{\mathcal{L}_D} \mathbf{D}\phi \Leftrightarrow \exists \Gamma' \subseteq \Gamma : (\Gamma' \not\models \bot \text{ and } \Gamma' \models \phi) \\ &\langle \Sigma, \Gamma, \mathcal{R}_D \rangle \models_{\mathcal{L}_D} \neg\kappa \Leftrightarrow \langle \Sigma, \Gamma, \mathcal{R}_D \rangle \not\models_{\mathcal{L}_D} \kappa \\ &\langle \Sigma, \Gamma, \mathcal{R}_D \rangle \models_{\mathcal{L}_D} \kappa_1 \wedge \kappa_2 \Leftrightarrow \langle \Sigma, \Gamma, \mathcal{R}_D \rangle \models_{\mathcal{L}_D} \kappa_1 \text{ and } \langle \Sigma, \Gamma, \mathcal{R}_D \rangle \models_{\mathcal{L}_D} \kappa_2 \end{aligned}$$

We expect a rational agent to try and manipulate its surrounding environment to fulfill its goals. In general, given a planning problem $\mathcal{P}$, not all goals can be fulfilled. For example, if in the description of the problem there is no action whose list of effects includes a goal, that goal will not be feasible. Hence, we assume a $\mathcal{P}$-dependent function $\mathcal{F}_\mathcal{P}$ that, given a belief base $\Sigma$ and a goal set $\Gamma^*$, returns $\top$ if $\Gamma^*$ is feasible and $\bot$ otherwise.

Function $\mathcal{F}_\mathcal{P}$ obeys the following axioms:

1. for all $\Sigma$, $\mathcal{F}_{\mathcal{P}}(\Sigma, \emptyset) = \top$ (an empty set of goals is always feasible);
2. for all $\Sigma$, $\Gamma_1^*$, $\Gamma_2^*$, if $\Gamma_1^* \subseteq \Gamma_2^*$,

$$\mathcal{F}_{\mathcal{P}}(\Sigma, \Gamma_1^*) = \bot \Rightarrow \mathcal{F}_{\mathcal{P}}(\Sigma, \Gamma_2^*) = \bot,$$
$$\mathcal{F}_{\mathcal{P}}(\Sigma, \Gamma_2^*) = \top \Rightarrow \mathcal{F}_{\mathcal{P}}(\Sigma, \Gamma_1^*) = \top$$

(goal set feasibility is monotonous).

In the example, where $\Sigma = \{\neg(p \wedge \neg u), \neg(u \wedge \neg r), \neg(s \wedge \neg u)\}$,

$$\mathcal{F}_{\mathcal{P}}(\Sigma, \emptyset) = \top,$$
$$\mathcal{F}_{\mathcal{P}}(\Sigma, \{r\}) = \top,$$
$$\mathcal{F}_{\mathcal{P}}(\Sigma, \{\neg s\}) = \top,$$
$$\mathcal{F}_{\mathcal{P}}(\Sigma, \{r, \neg s\}) = \top;$$

however, if we had $\Sigma' = \{p \wedge \neg r, \neg(u \wedge \neg r), \neg(s \wedge \neg u)\}$,

$$\mathcal{F}_{\mathcal{P}}(\Sigma', \emptyset) = \top,$$
$$\mathcal{F}_{\mathcal{P}}(\Sigma', \{r\}) = \bot,$$
$$\mathcal{F}_{\mathcal{P}}(\Sigma', \{\neg s\}) = \top,$$
$$\mathcal{F}_{\mathcal{P}}(\Sigma', \{r, \neg s\}) = \bot.$$

**Definition 7 (Feasible Candidate Goal Set)** *A candidate goal set $\Gamma^*$ is feasible for a planning problem $\mathcal{P}$ if and only if $\mathcal{F}_{\mathcal{P}}(\Sigma, \Gamma^*) = \top$.*

### 3.4 Comparing Candidate Goals and Sets of Candidate Goals

In this section we define one possible way in which an agent can choose among different sets of candidate goals. The particular choice made in this section is meant to illustrate goal comparison in the agent theory. If this particular way is replaced by another one, then still the general problem on choosing beliefs holds, and our solution can be applied.

¿From a desire base $\Gamma$, several candidate goal sets $\Gamma_i^*$, $1 \leq i \leq n$, may be derived. How can an agent choose among all the possible $\Gamma_i^*$? It is unrealistic to assume that for a rational agent all goals have the same importance. Therefore, we use the notion of expected pay-off to represent how relevant each goal is for the agent. The idea is that a rational agent tries to choose a set of goals which, first of all, is feasible and, secondly, gives the highest pay-off.

A pay-off function is a function $f : \mathcal{L} \rightarrow \mathbb{R}$ which associates a real value (the pay-off) to every formula in $\mathcal{L}$. Given $\phi \in \mathcal{L}$, $f(\phi)$ is the pay-off the agent would receive if $\phi$ were true. For all $\phi \in \Gamma$, we assume that $f(\phi) > 0$. In other words, a rational agent cannot desire something that, if realized, would bring no benefit.

One problem with pay-offs is that an agent may not always be able to attach a precise numerical value to its desires. An alternative approach would be to assume a total order over an agent's desires. In either case, we can define a total

order $\succeq$ between goals, such that, for all $\phi_1, \phi_2 \in \mathcal{L}$, $\phi_1 \succeq \phi_2$ iff $f(\phi_1) \geq f(\phi_2)$, or the agent desires $\phi_1$ at least as much as it desires $\phi_2$.

In the example, we could express the fact you want most of all to be reelected and, if possible, would rather not increase public spending, by defining $f(r) > f(\neg s)$, e.g., $f(r) = 100$, $f(\neg s) = 10$. In case you find it unnatural to assign arbitrary numerical values to these payoffs, you could just use a total order $\succeq$, and define $r \succ \neg s$.

The $\succeq$ relation can be extended from candidate goals to sets of candidate goals. For the qualitative ordering, we have that a goal set $\Gamma_1^*$ is preferred to another one $\Gamma_2^*$ if, considering only the goals occurring in one of the sets, the best goals are in $\Gamma_1^*$ or the worst goals are in $\Gamma_2^*$. Note that $\succeq$ is connected and therefore a total pre-order, i.e., we always have $\Gamma_1^* \succeq \Gamma_2^*$ or $\Gamma_2^* \succeq \Gamma_1^*$.

**Definition 8 (Preference between Sets of Candidate Goals)** *Given two candidate goal sets $\Gamma_1^*$ and $\Gamma_2^*$:*

- *We say that $\Gamma_1^*$ is at least as preferred as $\Gamma_2^*$ (denoted by $\Gamma_1^* \succeq \Gamma_2^*$):*

$$\sum_{\phi \in \Gamma_1^*} f(\phi) \geq \sum_{\phi \in \Gamma_2^*} f(\phi)$$

  *if the pay-offs are defined.*
- *If a preference relation over candidate goals is given, let $\Gamma_1' = \Gamma_1^* \setminus \Gamma_2^*$ and $\Gamma_2' = \Gamma_2^* \setminus \Gamma_1^*$. We then say that $\Gamma_1^* \succeq \Gamma_2^*$ iff one of the following two conditions is satisfied:*
  1. *$\forall \phi_2 \in \Gamma_2'$, $\exists \phi_1 \in \Gamma_1'$, s.t. $\phi_1 \succeq \phi_2$;*
  2. *$\forall \phi_1 \in \Gamma_1'$, $\exists \phi_2 \in \Gamma_2'$, such that $\phi_1 \succeq \phi_2$.*

In the example, it is easy to verify that the following relation holds in either cases (if a payoff function is defined or if a preference order is used):

$$\{r, \neg s\} \succ \{r\} \succ \{\neg s\} \succ \emptyset.$$

### 3.5 Defining the Goal Set Selection Function

In general, given a set of desires $\Gamma$, there may be many possible candidate goal sets. A rational agent in state $\mathcal{S} = \langle \Sigma, \Gamma, \mathcal{R}_D \rangle$ will select as the set of goals it wants to pursue one precise candidate goal set $\Gamma^*$ among the most preferred feasible candidate goal sets, which depends on $\mathcal{S}$.

Let us call $G$ the function which maps a state $\mathcal{S}$ into the goal set selected by a rational agent in state $\mathcal{S}$: $\Gamma^* = G(\mathcal{S})$.

In the example above, $G(\mathcal{S}) = \{r, \neg s\}$, because $\{r, \neg s\}$ is the most preferrable among the feasible goal sets.

## 4 Situating the Problem: Indeterministic Belief Change

> "Most models of belief change are deterministic in the sense that given a belief set and an input, the resulting belief set is well-determined. There is no scope for chance in selecting the new belief set. Clearly, this is not a realistic feature, but it makes the models much simpler and easier to handle, not least from a computational point of view. In indeterministic belief change, the subjection of a specified belief set to a specified input has more than one admissible outcome.
> Indeterministic operators can be constructed as sets of deterministic operations. Hence, given $n$ deterministic revision operators $*_1, *_2, \ldots, *_n$, $* = \{*_1, *_2, \ldots, *_n\}$ can be used as an indeterministic operator." [15]

Let us consider a belief set $\Sigma$ and a new belief $\beta$. The revision of $\Sigma$ in light of a new belief $\beta$ is simply:

$$\Sigma * \beta \in \{\Sigma *_1 \beta, \Sigma *_2 \beta, \ldots \Sigma *_n \beta\}. \tag{1}$$

More precisely, revising the belief set $\Sigma$ with the indeterministic operator $*$ in light of new belief $\beta$ leads to one of the $n$ belief revision results:

$$\Sigma * \beta \in \{\Sigma^1_\beta, \Sigma^2_\beta, \ldots \Sigma^n_\beta\}, \tag{2}$$

where $\Sigma^i_\beta$ is the $i$th possible belief revision result.

Applying operator $*$ is then equivalent to applying one of the virtual operators $*_i$ contained in its definition. While the rationality of an agent does not suggest any criterion to prefer one revision over the others, a defining feature of a CW agent is that it will choose which revision to adopt based on the consequence of that choice. One important consequence is the set of goals the agent will decide to pursue.

In the example, $\beta = \mathbf{B}(p \wedge \neg r)$, and

$$\Sigma * \beta \in \left\{ \begin{array}{l} \Sigma^1_\beta = \{p \wedge \neg r, \neg(u \wedge \neg r), \neg(s \wedge \neg u)\}, \\ \Sigma^2_\beta = \{p \wedge \neg r, \neg(p \wedge \neg u), \neg(s \wedge \neg u)\} \end{array} \right\}. \tag{3}$$

In the next sections we propose some possible ways to tackle the problem of choosing one of the revision options among the different available.

## 5 Belief Revision as a Decision Problem

By considering an indeterministic belief revision, we admit $\Sigma * \beta$ to have more than one possible result. In this case, the agent must select one among all possible revisions. Many criteria can be considered for selection. One of the criteria is to choose the belief revision operator for which the goal set selection function returns the most preferrable goal set. In other words, selecting the revision amounts to solve an optimization problem.

### 5.1 Indeterministic State Change

The indeterminism of belief revision influences the desire-updating process. In fact, the belief revision operator is just a part of the state-change operator, which is indeterministic as well, as a consequence of the indeterminism of belief revision. Therefore,

$$\mathcal{S}_\beta \in \{\mathcal{S}^1_\beta, \mathcal{S}^2_\beta, \ldots, \mathcal{S}^n_\beta\}, \tag{4}$$

where $\mathcal{S}^i_\beta = \langle \Sigma^i_\beta, \Gamma^i_\beta, \mathcal{R}_D \rangle$.

Which goal set is selected by an agent depends on $G$:

$$G(\mathcal{S}_\beta) \in \{G(\mathcal{S}^1_\beta), G(\mathcal{S}^2_\beta), \ldots, G(\mathcal{S}^n_\beta)\}. \tag{5}$$

In the example,

$$G(\mathcal{S}_\beta) \in \{G(\mathcal{S}^1_\beta), G(\mathcal{S}^2_\beta)),$$

where $G(\mathcal{S}^1_\beta) = \{r\}$ and $G(\mathcal{S}^2_\beta) = \{\neg s\}$. The following table summarizes the possibilities you may face when choosing between the two alternative revisions.

| reality $\rightarrow$<br>$\downarrow$ beliefs | $\not\models p \supset u$<br>$\models u \supset r$ | $\models p \supset u$<br>$\not\models u \supset r$ |
|---|---|---|
| $\Sigma^1_\beta$<br>plan: increase<br>public spending | $r$ is achieved<br>$\neg s$ is not achieved<br>pay-off $= f(r)$ | no desire<br>is achieved<br>pay-off $= 0$ |
| $\Sigma^2_\beta$<br><br>plan: do nothing | $r$ is not achieved<br>$\neg s$ is achieved<br>pay-off $= f(\neg s)$ | |

A traditional rational agent could not choose one of the $G(\mathcal{S}^i_\beta)$ because they are incomparable. Now, for a CW agent,

$$G(\mathcal{S}_\beta) \in \mathrm{PS}\{G(\mathcal{S}^1_\beta), G(\mathcal{S}^2_\beta), \ldots, G(\mathcal{S}^n_\beta)\}, \tag{6}$$

where $\mathrm{PS}(S)$ denotes the preferred set of $S$ defined as follows:

**Definition 9 (Preferred Set** PS**)** *Given two sets $S$ and $X$ such that $S \subseteq X$, and given a preference relation $\succeq$ over $X$, the preferred set of $S$ is*

$$\mathrm{PS}(S) = \{x \in S : \forall x' \in S, x \succeq x'\}. \tag{7}$$

Since in the example $G(\mathcal{S}^1_\beta) = \{r\}$ and $G(\mathcal{S}^2_\beta) = \{\neg s\}$, and $\{r\} \succ \{\neg s\}$,

$$\mathrm{PS}\{G(\mathcal{S}^1_\beta), G(\mathcal{S}^2_\beta)\} = \mathrm{PS}\{\{r\}, \{\neg s\}\} = \{\{r\}\}.$$

Therefore, a CW agent should choose revision $\Sigma^1_\beta$, because it is the only revision whereby you could possibly end up being re-elected, which is what you desire most. This is in agreement with the intuition underlying the motivating example.

### 5.2 Choosing a Revision

As long as different revisions lead to distinct goal sets with different degrees of preference, it is clear what revision a CW agent should choose. However, we can distinguish two situations in which the choice is less trivial:

- there is just one preferred goal set $\Gamma^*$, but more than one alternative options lead to $\Gamma^*$;
- there is no unique preferred goal set; that is, there are different goal sets $\Gamma^*_1, \ldots, \Gamma^*_m$, none of which is strictly preferred to the others, i.e., for all $i, j \in \{1, \ldots, m\}$, $\Gamma^*_i \succeq \Gamma^*_j$.

In these cases, some alternative belief revisions lead to equally preferred goal sets, and such revisions may be regarded as equivalent.

**Definition 10 (Equivalence between Belief Revision Candidates)** *A belief revision candidate $\Sigma^1_\beta$ is equivalent to another belief revision candidate $\Sigma^2_\beta$ (denoted by $\Sigma^1_\beta \approx \Sigma^2_\beta$), if and only if $G(\mathcal{S}^1_\beta) \succeq G(\mathcal{S}^2_\beta)$ and $G(\mathcal{S}^2_\beta) \succeq G(\mathcal{S}^1_\beta)$.*

It is easy to verify that $\approx$ is a standard equivalence relation, i.e., reflexive, symmetric, and transitive.

The choice of which revision outcome to adopt may thus be deterministic or indeterministic. It is indeterministic in the two cases presented above. More precisely, the choice depends on the preference relations over the goal sets, which determines the equivalence between revision candidates:

- if $\|\mathrm{PS}\{G(\mathcal{S}^1_\beta), G(\mathcal{S}^2_\beta), \ldots, G(\mathcal{S}^n_\beta)\}\| = 1$, i.e., the equivalent class of a preferred belief revision is a singleton and, if there is no $i, j$ such that $G(\mathcal{S}^i_\beta) = G(\mathcal{S}^j_\beta)$, the choice of the belief operator is obviously deterministic;
- if $\|\mathrm{PS}\{G(\mathcal{S}^1_\beta), G(\mathcal{S}^2_\beta), \ldots, G(\mathcal{S}^n_\beta)\}\| = 1$, and there is at least a couple $i, j$ such that $G(\mathcal{S}^i_\beta) = G(\mathcal{S}^j_\beta)$, the choice may be indeterministic, if two or more distinct revisions lead to one and the most preferred goal set, but also indifferent in practice.
- if $\|\mathrm{PS}\{G(\mathcal{S}^1_\beta), G(\mathcal{S}^2_\beta), \ldots, G(\mathcal{S}^n_\beta)\}\| > 1$, the choice is indeterministic;

It is important to notice that an agent that has to choose between $\Sigma^i_\beta$ and $\Sigma^j_\beta$ which lead to the same goal set (as in the second case above) is in a different situation than an agent who has to randomly choose among a number of competing revisions (as in the third case above). In the second case, whatever the agent's choice is, the goals are the same; in the third case, depending on the agent's choice, the goals the agent will pursue may vary. In general, a random choice is hardly a rational option. But, when an agent is in the second situation, it knows that, no matter which revision it chooses, the outcome does not change. In such a context, a random choice becomes a rational option.

**Proposition 1** *Let $*$ be an indeterministic belief operator, and $n$ be the number of possible belief revisions candidate. We have:*

$$1 \leq \|\mathrm{PS}\{G(\mathcal{S}^1_\beta), G(\mathcal{S}^2_\beta), \ldots, G(\mathcal{S}^n_\beta)\}\| \leq n.$$

### 5.3 Conditions for Determinism of a CW Agent

Traditional indeterministic belief revision approaches allow for the result of belief revision to be indeterminate in the sense that there may be many possible revision alternatives that are equally rational. Our proposal builds on the idea that what an agent wishes to achieve can play a role in the choice of which beliefs to reject and which beliefs to retain. The example we have been using in this paper also tries to capture the intuition that an agent who behaves in this manner is rational. Our richer model can distinguish one revision alternative from the other depending on the effect that each option has on the agent's goal set. Hence, under certain conditions, the choice among several revision alternatives can be reduced to one. This is what we want to investigate now, that is we want to investigate the conditions under which a revision for a CW agent is deterministic even if an indeterministic revision operator is used, i.e., $\|\mathrm{PS}\{G(\mathcal{S}^i_\beta)\}_{i=1,\ldots}\| = 1$ and, for all $i, j$, $G(\mathcal{S}^i_\beta) \neq G(\mathcal{S}^j_\beta)$. Determinism may be desirable, for instance, in agent programming, in those cases where predictability of the agent's behaviour is a requirement.

**Observation 1** *$\Sigma * \beta$ is deterministic in state $\mathcal{S} = \langle \Sigma, \Gamma, \mathcal{R}_D \rangle$, iff no two alternative revisions are equivalent, i.e., for all $i, j$, $\Sigma^i_\beta \not\approx \Sigma^j_\beta$.*

**Proposition 2** *A sufficient condition for no two alternative revisions, $\Sigma^i_\beta$ and $\Sigma^j_\beta$, being equivalent is that*

1. *for all $i, j$, $G(\mathcal{S}^i_\beta) \neq G(\mathcal{S}^j_\beta)$;*
2. *(a) if pay-offs are defined, for all $i, j$, $\sum_{\phi \in G(\mathcal{S}^i_\beta)} f(\phi) \neq \sum_{\phi \in G(\mathcal{S}^j_\beta)} f(\phi)$;*
   *(b) if pay-offs are not defined, the preference relation on goals is strict, i.e., for all $\phi, \phi' \in G(\mathcal{S}_\beta)$, $\phi \neq \phi'$, $\phi \succeq \phi' \Rightarrow \phi' \not\succeq \phi$.*

**Proof:** If pay-offs are defined, from Hypothesis 1 and 2a, by applying Definition 8, we obtain that either $(G(\mathcal{S}^i_\beta) \succeq G(\mathcal{S}^j_\beta)$ and $G(\mathcal{S}^j_\beta) \not\succeq G(\mathcal{S}^i_\beta))$ or $(G(\mathcal{S}^j_\beta) \succeq G(\mathcal{S}^i_\beta)$ and $G(\mathcal{S}^i_\beta) \not\succeq G(\mathcal{S}^j_\beta))$. Therefore, $\Sigma^i_\beta \not\approx \Sigma^j_\beta$.

If pay-offs are not defined, from Hypothesis 1 and 2b, by applying Definition 8, we obtain again $\Sigma^i_\beta \not\approx \Sigma^j_\beta$.

Therefore, no two alternative revisions can be equivalent. □

**Proposition 3** *If the pay-off function $f$ is such that, for all $\phi \in \mathcal{L}$, for all $\Psi \subseteq \mathcal{L} \setminus \{\phi\}$,*

$$f(\phi) \neq \sum_{\psi \in \Psi} f(\psi), \tag{8}$$

*Condition 2a of Proposition 2 always holds.*

The proof is trivial.

One might wonder how difficult it is to design a pay-off function that satisfies Inequality 8. The answer is, quite easy.

**Proposition 4** *Given a rational, injective pay-off function $f$, there exists another pay-off function $\hat{f}$ such that*

1. *$\hat{f}$ satisfies Inequality 8;*
2. *for any desired $\delta$, for all $\phi \in \mathcal{L}$, $|f(\phi) - \hat{f}(\phi)| < \delta$;*
3. *for all $\phi, \psi \in \mathcal{L}$, $f(\phi) > f(\psi) \Leftrightarrow \hat{f}(\phi) > \hat{f}(\psi)$.*

**Proof:** Since $f$ is rational, there exists $u \in \mathbb{R}$ such that, for all $\phi \in \mathcal{L}$, $f(\phi) = n$ for some integer $n$. Let

$$\epsilon_0 < \min\{\delta, u, \min_{\phi \in \mathcal{L}} f(\phi)\}.$$

We define a sequence $\{\epsilon_i\}_{i=0,\ldots}$ such that $\epsilon_{i+1} = \epsilon_i/2$. It easy to verify that no element $\epsilon_i$ can be obtained as a sum of a finite number of other elements $\epsilon_j$, with $i \neq j$. Now, let $\phi_1$, $\phi_2$, ... an effective enumeration of all formulas in $\mathcal{L}$ (such enumeration, which needs not be finite, exists for all recursively enumerable languages); pay-off function $\hat{f}$ may be defined, for all $i = 1, 2, \ldots$, as

$$\hat{f}(\phi_i) = f(\phi_i) + \epsilon_i.$$

Function $\hat{f}$ satisfies the three conditions of the thesis. □

## 6 Related work

### 6.1 Goal Change

In this paper we do not explain the process of goal generation and revision, i.e., we are not interested in how new goals arise in the light of new beliefs or desires. That aspect is considered, for example, in [5,6], where an approach has been proposed to dynamically construct the goal set to be pursued by a rational agent, by considering changes in its mental state. More precisely, the autors propose a general framework based on classical propositional logic, to represent changes in the mental state of the agent after the acquisition of new information and/or after the arising of new desires.

An important point of this framework, which distinguishes it from the framework used in this paper, is that the two aspects of how goals are selected by an agent and how the selected goals are achieved are not conceptually separated: this means, the goal selection mechanics depend on the planning process and then interactions between these two aspects are a part of the goal generation/revision process.

### 6.2 BOID

The BOID architecture [2] extends a classical planner with a component for goal generation. In this goal generation component, there are subcomponents for beliefs, obligations, intentions and desires [4]. The interaction among these

subcomponents is studied using a qualitative decision theory [3,12] and qualitative game theory [10] based on extensions of input/output logic [17,18,1]. using merging operators [9], as an extension of the 3APL programming language [11], and using defeasible logic [7]. Though in all of these approaches the relation between beliefs and goals plays a central role, in these papers the impact of goals on the choice among belief sets has not been studied.

### 6.3 Preference over Beliefs

Doyle suggests to have a preference order over belief sets [13]. We have however an indirect link from belief sets to feasible goals, and a preference order over these goals; and from these preferences over goals, we again derive the preferences over belief sets. Therefore, if one wanted to accept Doyle's suggestion, our work could be regarded as a method for deriving a rationally justified preference order over belief sets.

## 7 Conclusions

We have presented some preliminary ideas for a new approach aiming at resolving indeterminism in belief revision. The framework has been inspired by the concept of *conventional wisdom*, introduced by economist John Kenneth Galbraith. Revising a belief base with an indeterministic operator in light of a new belief leads to more than one possible revisions. In this case, a traditional rational agent would not be able to choose among the possible revision candidates. The idea we started to develop is that the agent, in this case, may evaluate the effects that the different revision options have on its goals. Therefore, it could choose a revision which maximizes its goals. In other words, selecting the revision would amount to solving an optimization problem. Finally, fundamental definitions and properties of the belief revision mechanisms have been given.

Some topics for further research:

1. We would like to add a function $V(S, G)$ that returns for a state together with the feasible goals, the goals which also have been generated but which are not feasible. Then we can define preferences not only over feasible goals as now, but also over unfeasible ones.
   Before doing so, we have to be clear about what we think goals are: achievement goals on events/punctual or maintenance goals on states/continuant? In other words: if goals are events, then once you achieve them, you can forget about them, like when you shoot at a target and you hit it; but if goals can be states, like "staying alive", this does not hold anymore: it isn't because you believe you're alive that you don't want to stay alive anymore.
2. Assume that we have conflict between $\mathbf{B}p$ and $\mathbf{B}\neg p$, and we can choose either one of these beliefs, or none at all. What can we say about this situation? Should we be more adventurous by believing either $\mathbf{B}p$ or $\mathbf{B}\neg p$ rather than believing nothing, and if so, under which conditions? Consider for example

the principle of goal generation that more information about beliefs leads to more goals (monotony in beliefs for practical reasoning rules). In that case, under suitable conditions, we can probably prove that we can ignore the choice in which we do not believe anything.
3. Another open point of this work concerns the situation in which even if we you consider preferences among goal sets, this would not be enough for determining the belief revision to be adopted – belief revision process remains indeterministic. In this case, it is necessary to provide a framework which deals with this situation. One possibility would be to keep revision options open waiting for a new input which help us to choose the more convenient revision.
4. We can formalize the present model as an abstraction of a more general model of decision making (e.g. taking inspiration from decision trees, Savage style decision theory, action logics etc.) and consider the rationality of our CW agent in this more general theory.

## References


1. Boella, G., Hulstijn, J. and van der Torre, L., "Interaction in Normative Multi-Agent Systems", *Electronic Notes in Theoretical Computer Science*, **141(5)**, 2005, 135–162.
2. Broersen, J., Dastani, M., Hulstijn, J. and van der Torre, L., "Goal Generation in the BOID Architecture", *Cognitive Science Quarterly Journal*, **2**(3–4), 2002, 428–447.
3. Broersen, J., Dastani, M. and van der Torre, L., "Realistic Desires", *Journal of Applied Non-Classical Logics*, **12(2)**, 2002, 287–308.
4. Broersen, J., Dastani, M. and van der Torre, L., "Beliefs, Obligations, Intentions and Desires as components in an agent architecture", *International Journal of Intelligent Systems*, **20:9**, 2005, 893–919.
5. da Costa Pereira, C. and Tettamanzi, A., "Towards a Framework for Goal Revision", in: Pierre-Yves Schobbens, W. V. and Schwanen, G. (eds.), *BNAIC-06, Proceedings of the 18th Belgium-Netherlands Conference on Artificial Intelligence*, Namur, Belgium: University of Namur, 2006, 99–106.
6. da Costa Pereira, C., Tettamanzi, A. and Amgoud, L., "Goal Revision for a Rational Agent", in: Brewka, G., Coradeschi, S., Perini, A. and Traverso, P. (eds.), *ECAI 2006, Proceedings of the 17th European Conference on Artificial Intelligence*, Riva del Garda, Italy: IOS Press, 2006, 747–748.
7. Dastani, M., Governatori, G., Rotolo, A. and van der Torre, L., "Programming Cognitive Agents in Defeasible Logic", in: *Proceedings LPAR'05*, LNCS, Springer, 2005.
8. Dastani, M., Hulstijn, J. and van der Torre, L., "How to decide what to do?", *European Journal of Operational Research*, **160**(3), 2005, 762–784.
9. Dastani, M. and van der Torre, L., "Specifying the Merging of Desires into Goals in the Context of Beliefs", in: *Proceedings of The First Eurasian Conference on Advances in Information and Communication Technology (EurAsia ICT 2002)*, LNCS 2510, Springer, 2002, 824–831.
10. Dastani, M. and van der Torre, L., "Games for Cognitive Agents", in: *Proceedings of JELIA04*, LNAI 3229, 2004, 5–17.

11. Dastani, M. and van der Torre, L., "Programming BOID Agents: a deliberation language for conflicts between mental attitudes and plans", in: *Proceedings of the Third International Joint Conference on Autonomous Agents and Multi Agent Systems (AAMAS'04)*, 2004, 706–713.
12. Dastani, M. and van der Torre, L., "What is a normative goal? Towards Goal-based Normative Agent Architectures Regulated Agent-Based Systems", LNAI 2934, Springer, 2004, 210–227.
13. Doyle, J., "Rational Belief Revision", in: Allen, J. F., Fikes, R. and Sandewall, E. (eds.), *KR'91: Principles of Knowledge Representation and Reasoning*, San Mateo, California: Morgan Kaufmann, 1991, 163–174.
14. Galbraith, J. K., *The Affluent Society*, Boston: Houghton Mifflin, 1958.
15. Hansson, S. O., "Logic of Belief Revision", in: Zalta, E. N. (ed.), *The Stanford Encyclopedia of Philosophy*, Summer 2006.
16. Lindström, S. and Rabinowicz, W., "Epistemic entrenchment with incomparabilities and relational belief revision", in: Fuhrmann, A. and Morreau, M. (eds.), *The Logic of Theory Change*, 93–126.
17. Makinson, D. and van der Torre, L., "Input-output logics", *Journal of Philosophical Logic*, **29**, 2000, 383–408.
18. Makinson, D. and van der Torre, L., "Constraints for input-output logics", *Journal of Philosophical Logic*, **30(2)**, 2001, 155–185.
19. Olsson, E. J., "Lindström and Rabinowicz on relational belief revision", in: T. Ronnow-Rasmussen, J. J., B. Petersson and Egonsson, D. (eds.), *Hommage à Wlodek. Philosophical Papers Dedicated to Wlodek Rabinowicz*, 2007.
20. van Riemsdijk, M. B., *Cognitive Agent Programming: A Semantic Approach*, Ph.D. thesis, University of Utrecht, 2006.
21. Smith, D. E., "Choosing Objectives in Over-Subscription Planning.", in: Zilberstein, S., Koehler, J. and Koenig, S. (eds.), *Proceedings of the Fourteenth International Conference on Automated Planning and Scheduling (ICAPS 2004)*, Whistler, British Columbia, Canada: AAAI, 2004, 393–401.

# Control Patterns in a Health Care Network

Vera Kartseva[1], Joris Hulstijn[1], Jaap Gordijn[2] and Yao-Hua Tan[1]

[1] Faculty of Economics and Business Administration, Vrije Universiteit, Amsterdam
De Boelelaan 1105, 1081 HV Amsterdam
{ vkartseva, jhulstijn, ytan}@feweb.vu.nl
[2] Faculty of Sciences, Vrije Universiteit, Amsterdam
de Boelelaan 1081a 1081 HV Amsterdam
gordijn@few@vu.nl

**Abstract.** In this paper we present *control patterns* for the analysis and design of administrative control mechanisms in a network organization. A control pattern is a description of a generic and reusable control mechanism that solves a specific control problem, to be selected on the basis of the context. To represent the context and solution, we analyze a network organization as a set of actors who transfer objects of economic value. The usefulness and adequacy of the control patterns is demonstrated by a case study of the governance and control mechanisms of the Dutch public health insurance network for exceptional medical expenses (AWBZ).

**Keywords.** governance and control, network organizations, value modeling

## 1 Introduction

Multi-agent systems are computer systems which are composed of a number of autonomous agents, i.e., pieces of software, interacting to achieve the global systems objectives. Applications can for example be found in transport logistics [9], manufacturing scheduling [38], and social simulation [35]. Increasingly, developers of multi-agent systems have turned to concepts from the social sciences like organization structures and norms, as a guideline for design, e.g. [48][29]. Because of the desired global system objective, the autonomy of the individual agents must generally be restricted: the decisions of agents should be bound by rules, regulations and norms. Especially in an open environment, in which heterogeneous agents may freely enter, leave and interact, a multi-agent system becomes a *normative* multi-agent system [13]. Normative multi-agent 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" [23].

Generally, there are two good reasons to employ multi-agent systems: (1) as a paradigm for software engineering, or (2) as a conceptual tool for understanding human societies [47, p.7]. In the first case, multi-agent techniques are applied, because the

Dagstuhl Seminar Proceedings 07122
Normative Multi-agent Systems
http://drops.dagstuhl.de/opus/volltexte/2007/915

*solution* to some automation problem is inherently distributed. This is for example the case in manufacturing scheduling [38]. A centralized solution, as in operations research, would be possible, but would be less robust, and sometimes less effective or efficient. In the second case, multi-agent techniques are applied, because the *application problem* is inherently distributed, and a centralized solution is not feasible or legally possible. Consider for example a work-flow system for criminal investigations, in which parties with different legal rights (prosecution, defense, forensic expert), must or may not have access to the documents at certain stages of the proceedings.

In this paper, we provide a case study of the second type of application: the health care sector. We model a complex network organization in which autonomous entities (government agencies, care providers, patients) collaborate to achieve a global objective. Although we do not strictly use the terminology and techniques of the multi-agent community – we use theories from management and accounting – we do believe that this work is relevant to multi-agent systems community. First, because it provides a set of *control patterns*, practical guidelines for the analysis and design of the implementation of norms and regulations in a network organization. Such control patterns may also be of use in the design of normative multi-agent systems. Second, because it deploys the notion of *economic value* from business modeling to motivate the regulations. The use of economic value allows specification of the business model of a network organization, at a relatively high level of abstraction. But to understand these contributions, we first have to explain the case study.

### 1.1 A Social Chart for Dementia Care

In the Netherlands, the health care sector is making a transition from a supply-driven structure, in which health care providers (e.g. hospitals) decide what health care services are delivered, towards a more demand-driven structure, in which the patient can select health care services (e.g. treatment, physiotherapy, domestic care) from different providers. Such a 'market for health care' requires that patients know enough about the available services and providers to make an informed choice. Currently, patients generally perceive the health care on offer as fragmented, and not fitted to their needs [33]. Few information is available about the services offered by different care providers, and about their quality of those services.

This research is carried out as part of the FRUX project [14]. Among other things, the project develops a *dynamic interactive social chart for dementia care* (DEMDISC): an interactive web-site that will provide an overview of the health care services available in a region, and provide personalized advice about possible combinations of health care services, so called *service bundles*. The Social Chart is designed for the relatives and informal care givers of patients with Alzheimer's disease (dementia). One of the aims is to develop a generic method for generating bundles of services, tailored to the specific needs of a user [2].

### 1.2 Control Patterns and Value Modeling

Our role in the project is to model the governance and control aspects of a complex information service like the Social Chart. Before starting a software requirements engi-

neering process, it is crucial to get an understanding of the business model that underlies the health care network. The health care sector is highly regulated. Moreover, the regulations that govern the health care system are subject to change. Because changes are the outcome of a political process with many stake holders, the way they develop can remain unclear. Moreover, when health care regulations change, the business opportunities for a system like the Social Chart may well change too. For these reasons it is important to develop a generic method for analyzing and developing governance and control aspects, and for discovering corresponding business opportunities. Important elements of such a method are

(i) a representation language and graphical modeling tool for the analysis and redesign of control procedures, based on accounting principles,

(ii) a set of general guidelines to assist in analyzing and redesigning control procedures,

(iii) a library of generic control mechanisms, that have been validated and can serve as 'best practices'.

Regarding (i), the $e^3$-*control* methodology provides a representation language and graphical modeling tool [21]. The $e^3$-*control* methodology is based on the $e^3$-*value* ontology, which analyzes network organizations in terms of the transfer of *economic value* between participants [21]. The resulting model is called a *value network*. In this paper we extend this work, and present a set of *control patterns*. A control pattern is a description of a generic and re-usable control mechanism that solves a control problem, to be selected on the basis of the context of application. A control problem is an identifiable risk for opportunistic action by one of the other network participants. So control patterns combine guidelines (ii), with best practices (iii). Control patterns are inspired by the design patterns approach, which was proposed in architecture [1] and is now very successful in software engineering. More recently, it has also been applied in the business domain, to administrative processes [36], organizational structures [11], and business process re-engineering [4].

The usefulness and adequacy of control patterns is validated on a case study of the value network for the Dutch public health insurance system AWBZ (Exceptional Medical Expenses Act). The governance and control aspects of this system are interesting, because it is funded by taxes, and lacks direct feedback on the quality of services. The system is undergoing changes, one of which is the introduction of a personal budget. This facilitates the development of a kind of market for care providers. Although an analysis approach based on economic value works well in commercial settings, one could question its suitability for the public sector. We are therefore especially interested in the applicability of control patterns in a highly regulated value network, which involves many public-private partnerships. See also [26].

The remainder of the paper is structured as follows. In Section 2 we present a definition, and our library of control patterns, explaining the underlying control theory. In Section 3 we apply the control patterns to the case study, reverse engineering the way in which the network may have developed.

## 2 Control Patterns

A sustainable network organization needs mechanisms to govern and control the interaction among network participants. In most cases, interaction is encoded in contractual arrangements, and implemented through procedures and regulations. But regulations can be violated. In the context of control theory, a network organization is therefore considered to be either in an *ideal situation*, in which no opportunistic behavior in the form of violations, errors, or fraud will occur, or in a *sub-ideal situation*, in which violations, errors, or fraud do occur. The terminology ideal – sub-ideal is taken from deontic logic [34]. The ideal situation is in some sense normative or obliged, or (alternatively) desired by the dominant actor in the network. Sub-ideal situations can be prevented, detected or corrected by administrative measures called *control mechanisms*. An example of a control mechanism is the three-way verification by the accounting department of a payment for a purchase, against the invoice and the inventory.

In the accounting literature, control mechanisms are typically analyzed from an operational or procedural perspective, with process models and flow charts [43][39]. In a network organization, however, the ideal situation is often determined by the business model of the network. Therefore, we need a form of business modeling, to analyze the reasons for implementing a control mechanism.

### 2.1 Business Modeling

There are several methodologies that address the design of business models for network organizations, like the Business Modelling Ontology [37], value webs [45], and $e^3$-*value* [21]. Of these, $e^3$-*value* is the only one with a formal semantics, and a specific focus on value transfers between enterprizes. The method is ontologically well-founded, and is supported by graphical modeling tools. We therefore apply the $e^3$-*value* ontology [21] for the description of so called ideal models, which express organizations that behave in compliance with the procedures and regulations. Sub-ideal models are expressed using $e^3$-*control*, a modification of the $e^3$-*value* ontology, used to describe opportunistic behavior of actors [21].

*Ideal Value Models* An $e^3$-*value* model provides a conceptual model of the value transfers in a business network, encoded in the $e^3$-*value* ontology [21]. The $e^3$-*value* constructs have a graphical notation. Figure 1(a) shows an example of a buyer who obtains goods from a seller and offers a payment in return. According to the law, the seller is obliged to pay value-added tax (VAT). This can be conceptualized by the following $e^3$-*value* constructs (in **bold**).

**Actors**, such as the buyer, seller, and the tax office are economically independent entities. Actors transfer **value objects** (payment, goods, VAT) by means of **value transfers**. For each value object, some actor should be willing to pay, which is shown by a **value interface**. A value interface models the *principle of economic reciprocity*: actors are only willing to transfer a value object, in return for some other value object. So only if you pay, can you obtain the goods and vice versa. A value interface consists of **value ports**, to represent that value objects are offered to and requested from the actor's

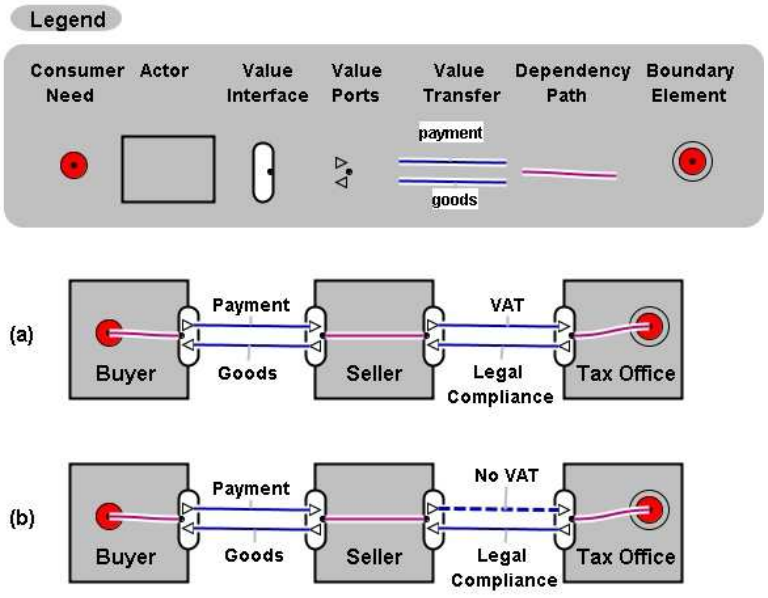



**Fig. 1.** Example of an $e^3$-*value* model of a purchase

environment. Actors may have a **consumer need**, which, following a path of **dependencies** will result in the transfer of value objects. Transfers are either dependent on other transfers, or lead to a **boundary element**. The $e^3$-*value* methodology also allows the designer to assign monetary values to value transfers in a spreadsheet and calculate profitability of actors in a network.

*Sub-Ideal Value Models* In $e^3$-*value* it is assumed that actors behave in an ideal way, meaning that all value transfers occur as prescribed. This implies, among other things, that actors respect the principle of economic reciprocity. But in reality, actors may commit fraud or make unintentional errors, e.g. actors will not pay, or not obtain the right goods. In $e^3$-*control* such sub-ideal value transfers are graphically represented by dashed arrows [21]. For example, Figure 1(b) shows a sub-ideal situation in which the seller does not pay VAT.

*Process Models* Value models consider the transfer of value objects, like money, goods or services. This is essentially the transfer of ownership rights [25]. But such transfer requires operational activities to be performed, by multiple actors, which can only be shown using a process model. Moreover, in value models the temporal order in which objects are transferred is abstracted over: it only represents that objects are transferred, but not in which order. The order in which activities take place, forms a crucial part of many control mechanisms. So in addition to value models we need process models to capture control aspects. We represent process aspects of control problems and their solutions by UML-activity diagrams [40].

*Information Models* Evidence documents, like receipts or tickets, form an important part of the administrative control mechanisms studied in this paper. Nowadays, documents are often certified files in a distributed information system, of which the paper document is only a trace. For an analysis of the structure and content of documents, UML class diagrams are a suitable representation format. Other issues are related to the management of information. Which party should store the documents? How is privacy, accuracy and reliability of the data preserved? For such issues, see Ronmey and Steinbart [39, ch8]. In this paper, we focus only on the procedural role of documents in a control mechanism.

### 2.2 Theoretical Framework

The control patterns proposed later in this section, are based on a combination of agency theory, control theory and ideas about trust.

*Agency Theory* A well known theory in sociology and management accounting is *agency theory*, also called *principal-agent* theory. See Eisenhardt [16] for a survey. Agency theory studies the relationship between two parties: the *principal*, who delegates some activity, and the *agent*, to whom the activity is delegated. The theory argues that if (1) the principal and the agent are utility maximizers with bounded rationality and (2) there is information asymmetry in favor of the agent, the agent may behave opportunistically. Agency theory distinguishes two types of opportunistic behavior.

The first type is caused by *hidden information*: the principal can not be sure that the agent accurately presents his ability to do the work. For example, a producer (agent) generally has better information about the product he is producing, than someone who wants to buy the product (principal). The generally accepted control mechanism against hidden information is *screening* [28]: the principal collects information about the reliability of the agent, before agreeing on a transaction.

The second type is caused by *hidden action*: the principal can not be sure whether the agent did his work according to the contract or not. For example, the producer may use low quality components to produce a product. As a result, the quality of the product is lower than agreed in the contract. The generally accepted control mechanisms against hidden action are *monitoring* the agent, the and *creation of incentives* to motivate the agent not to behave opportunistically [16].

*Control Theory* Control problems are typically identified by an analysis of risk indicators and threats discovered in an audit process. A control mechanism prescribes how to organize business processes in order to prevent, detect or reduce the risks posed by a control problem. Internal control theory is concerned with administrative and organizational measures inside an organization [43]. But in inter-organizational settings, risks related to the behavior of partners in a network, are mostly dealt with by contractual arrangements. So it is difficult to apply internal control guidelines directly. We also used the more formal work of Chen [10] and Bons et al. [8][7] on *inter-organizational trade procedures*. Chen concentrates on *detective controls*, as they are applied in purchasing and procurement. Bons et al also deal with forms of *preventative control*.

The distinction between detective and preventative control can be made clear by the well-known example of the Paris metro [18]. In many countries, the norm that one should not board a train without a ticket, is implemented by random checks. This is a form of detective control; it assumes that there are enough incentives for a passenger to avoid violation, given that the chance of being caught and the (social) sanction are large enough. By contrast, the Paris metro has an elaborate system of automatic gates, which makes it physically impossible to board a train without a ticket. This is a form of preventative control. In practice, the two types of control are often combined.

*Trust* A network organization can be interpreted as a number of binary value transactions between actors. When parties in a network do not have an existing business relationship, lack of trust is likely. Trust has been defined as

> "The willingness of a party to be vulnerable to the actions of another party based on the expectation that the other party will perform a particular action important to the trustor, irrespective of the ability to monitor or control that other party"[32, p.712].

Without prior trust, the party who invests in a transaction, called the *trustor*, is uncertain whether the other party, the *trustee*, will perform its part of the deal or will defect and behave opportunistically. The uncertainty of the trustor about possible opportunistic behavior also comes out in Gambetta's definition:

> "Trust is the subjective probability by which an individual $a$ expects that another individual $b$ performs a given action on which its welfare depends" [19].

However, trust does not have to depend on the trustor and trustee alone [44]. Institutional control measures can be used to guarantee performance according to contract. Legal provisions against fraud are an example of a detective kind of control mechanism. An example of a preventative control mechanism for international trade, is an Escrow service [22]. In such a scenario, the money of the buyer is deposited with a trusted third party, called Escrow agent. Only when the Escrow agent confirms to the seller that the money has been deposited, the goods are shipped. This solves the risk of the seller, that the buyer will or cannot pay. Only when the buyer has confirmed to the Escrow agent, that the goods have been delivered as agreed, the money is released and transferred to the seller, with a handsome fee subtracted. This solves the risk of the buyer, that the seller will not deliver as agreed. So in general, the purpose of inter-organizational control mechanisms is to reduce the uncertainty of the trustor, and provide enough guarantees for both parties to engage in a transaction. If we model the reasoning of the trustor, we get a kind of game theoretic reasoning, comparing a scenario in which a control mechanism applies, with a scenario in which there are no institutional control measures [5]. This kind of reasoning, about the risks of engaging in a transaction, can be applied from the perspective of both the buyer and the seller.

It is not a coincidence that the effectiveness of an Escrow service has been shown by game theoretic means [22]. The same kind of reasoning – using recursive models of the expected behavior of agents according to other agents – can also be modeled by the 'qualitative' game theory developed by Boella and Van der Torre [6]. Although we do not explicitly use this kind of recursive modeling, it is implicit in the way control patterns are to be applied.

### 2.3 Conceptual Framework

The three components of the underlying theory, agency theory, control theory and trust, can be combined. Based on this mixed background, we identify a vocabulary of terms (in **bold**), to be used in the definition of control patterns.

Consider a transaction scenario, as shown in Figure 2. There are two agents, called **primary actor** and **counter actor**. From a value perspective, we say that the primary actor transfers a value object, called **primary object**, to the counter actor, and the counter actor returns a value object called **counter object**. From a process perspective, such a transaction is modeled by two operational activities: the primary actor performs a **primary activity**, and the counter actor executes a **counter activity**, each resulting in the corresponding value transfers[3]. Figure 2 shows a value model of such a transaction on the left, and the corresponding process model on the right. The order in which the primary activity and counter activity occur is not specified. This is indicated by the UML notation for parallel execution (thick horizontal bar).

As in agency theory, in this scenario the primary actor has delegated some activity to the counter actor. So we could say that **primary actor = principal**, and **counter actor = agent**. The control risks of the transaction are generally assessed from the point of view of the primary actor, who – by definition – does not trust the counter actor, and must therefore design control mechanisms against possible sub-ideal behavior of the counter actor. So we could also say that **primary actor = trustor**, and **counter actor = trustee**.

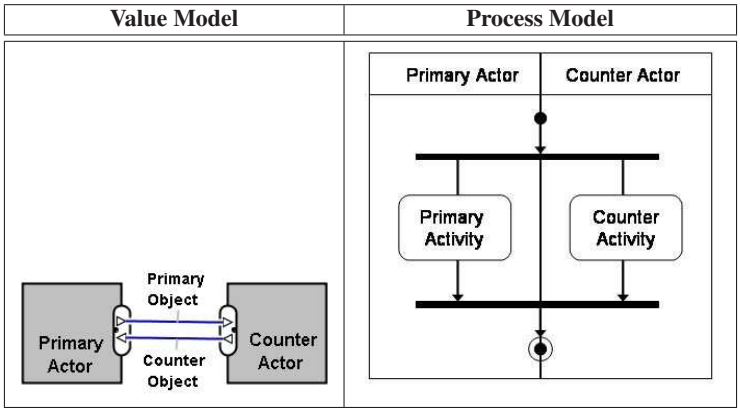



**Fig. 2.** Transaction Scenario, in which primary actor = principal = trustor, and counter actor = agent = trustee.

Regarding detective control, we need to make some more distinctions. Detective control [10] is generally concerned with *verification*: comparing the results of an op-

[3] The primary-counter terminology is based on Bons [7], and is inspired by contract law, which uses the phrases *primary obligation*, and *counter party*.

erational activity with some claim about its legitimacy, quality or quantity. The claim is represented by a *document to-be-verified*. One or more so called *supporting documents* represent evidence about the operational activity. The result of a verification is usually a decision to perform some action or not, or else an evidence document stating the decision. A template of a verification activity is shown in Figure 3. To simplify the diagrams, in the remainder of the paper, we will only show the positive outcome of a verification activity, since the negative outcome always leads to the end of the process. One way of obtaining evidence about an activity, is by a **witness** activity. When parties sign a contract, evidence of commitments is generated by **confirm**. For the transfer of a value object, we use activities **request** and **provide**. When the value object is a fee, we use the activity **pay**.

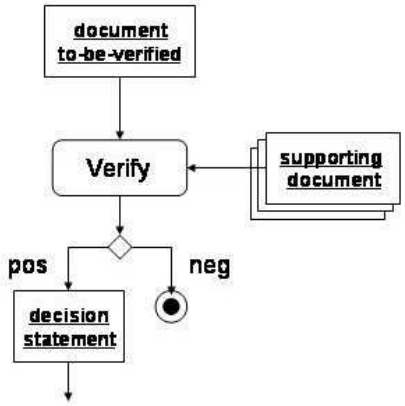



**Fig. 3.** Verification Activity

### 2.4 Pattern Definition

Control patterns are inspired by the use of design patterns in architecture [1] and software engineering [20]. The idea is to capture the 'best practices' about the design of buildings, software, and later also organizational structures [4][36], for different applications. Traditionally, a design pattern has four essential elements: pattern name, problem, solution, and consequences [20, p.3]. We have transferred the idea of using design patterns to the control domain. We separated the description of the context in which a pattern is to be applied, from the problem which motivates the selection of a pattern [11]. In our interpretation, the context describes the value network with the actors, their relationships, like trust, and the activities to be controlled. The problem specifies a risk to be detected or prevented by the solution of the pattern. The solution describes the value network and the corresponding process model, after implementing the control mechanism encoded in the pattern.

**Definition 1 (control pattern).** A *control pattern* is a description of a generic and reusable control mechanism for a recurring control problem, selected on the basis of the context. A control pattern consists of the following elements:

**name:** a descriptive name of the pattern, used to select patterns from a pattern library.
**context:** a description of the business network to be controlled, modeled from an ideal perspective, meaning that no one behaves opportunistically. The context is represented by a value model, and if needed for understanding the context, also by a process model.
**problem:** a statement of risks for opportunistic behavior. A control problem exists if there is some deviation of the prescribed transfers of economic value. So we model the problem by a sub-ideal value model, using sub-ideal value transfers. Again, if needed, we also use a process model.
**solution:** description of a control mechanism, to detect, prevent or correct the control problem.

The solution is described by both process models and value models. Table 1 contains an overview of our library of reusable control patterns. They were elicited using the PattCaR method [42]. Based on literature and case studies, we identified potential patterns. These were modeled with $e^3$*-control* and activity diagrams, and compared using a commonality-variability analysis. The resulting patterns are validated in case studies, one of which is presented here.

In addition we present some *organizational design patterns* listed in Table 2. Compare for example [31][11]. We use delegation of an activity to some external actor. For this pattern, we assume that the actor to whom the activity is delegated, is trusted by the primary actor, so no additional control measures are required. Otherwise, the usual control patterns apply. Decomposition of activities is typically used for efficiency. It is also needed to separate an activity into two parts for control reasons, as for example when a payment is made in two installments, one before and one after delivery (down payment).

### 2.5 Library of Control Patterns

We will now present a library of reusable control patterns. We illustrate the patterns by a simplified transaction scenario, in which a buyer (primary actor) has ordered some goods, and does not trust the seller (counter actor) to deliver. So in this case the primary activity is payment; the counter activity is delivery of the goods.

The **Execution Monitoring** pattern requires the primary actor to verify execution of the counter activity, *before* executing the primary activity [10][7]. In previous versions of this research [27], the pattern was therefore called Pre-execution. It roughly corresponds to the monitoring practice of agency theory [16]. The associated control problem for the primary actor is that without the verification, the primary actor is not certain whether the counter actor will execute the counter activity as agreed. In our example, the buyer is not certain about delivery. So the buyer will only pay for the goods, after having inspected that the right goods were delivered. When both actors apply this pattern, we get into conflict, and an intermediate solution like a down-payment must be applied [27].

| **Name** | **Control Problem** | **Solution** |
|---|---|---|
| Execution Monitoring | primary actor is not certain whether counter actor will execute the counter activity as agreed | verify counter activity, before executing primary activity |
| Execution Confirmation | counter actor may deny that primary activity was executed as agreed, and refuse to execute counter activity, or require compensation | require confirmation from counter actor that primary activity was executed as agreed |
| Commitment Confirmation | counter actor may deny to have made a commitment to primary actor | require confirmation of commitment from counter actor |
| Partner Screening | counter actor may not be a reliable partner to make commitments with before making any commitments | verify credentials of counter actor |
| Certification | counter actor may not be reliable to perform counter activities | require verification of past behavior of counter actor |

**Table 1.** Library of Control Patterns

| Name | Objective | Solution |
|---|---|---|
| Delegation | activity can not be effectively or efficiently done | delegate activity to a specialized trusted external actor |
| Decomposition | activity can not be effectively or efficiently done | decompose activity in parts, which can be effectively or efficiently carried out |

**Table 2.** Organizational Design patterns

The **Execution Confirmation** pattern requires the counter actor to confirm that the primary activity was performed as agreed [7]. Think of a receipt. The associated control problem is that, in case of a dispute, the primary actor will not have independent evidence to prove that the primary activity was executed as agreed. In our example, the buyer will require a quittance from the seller, as evidence of payment.

The **Commitment Confirmation** requires the counter-actor to confirm, i.e. provide documentary evidence, of the transaction commitment [7][46]. Normally, this is done by signing a contract. The associated control problem is that otherwise the counter actor may refuse to recognize that he made a commitment to the primary actor, and may therefore not execute the counter activity. In our example, the buyer will require a price quote or offer, which commits the seller to deliver at a certain price.

The **Partner Screening** pattern require the primary actor to collect evidence about the past conduct of the counter actor, and verify whether the past record and reputation of the counter actor conform to standards of trustworthiness, before making any commitments [28]. The underlying control problem is the risk of making a commitment to an unreliable partner. Often the research and verification are delegated to a specialized party. In our example, the buyer can have an agency like the Chamber of Commerce check the credentials of the seller, before making any commitments.

The **Certification** pattern establishes the authorization by the primary actor that the counter actor meets the requirements to be allowed to perform a counter activity of that type. This kind of regulatory control is can be implemented by different kinds of evidence documents, like certificates, licenses or accreditations. Certification is needed when the primary actor is at least partly responsible for the counter activity. The primary actor can be a regulatory body, or an actor who has delegated the execution of the counter activity to another actor. For example, companies offering Escrow services must be licensed and accredited by the financial regulatory bodies of the country in which they are based, such as the central bank.

In the Appendix, the patterns are summarized using the graphical notation discussed above.

### 2.6 How to apply the Control Patterns?

Patterns can be applied using the three steps of the $e^3$*-control* methodology [24]:

1. Define ideal value models
2. Identify control problems; model them in sub-ideal value models and sub-ideal process models
3. Select a pattern to fit the control problem; (re-)design control mechanisms, using the pattern.

To select a pattern from the patterns library, a control problem in a case description must be matched with the control problem in one of the patterns. First sub-ideal value and process models of the case are developed. Then, based on the value model, one can identify the primary actor, counter actor and the sub-ideal value transfer. The primary actor is the one who may expect sub-ideal behavior from the counter actor. After a pattern that matches the problem description is found, the process model is adjusted according to the pattern.

## 3 Validation In Health care

In this section, we show how the control patterns can be applied to a health care network. In particular, we focus on the fact that the case exemplifies a highly regulated value network that involves many private-public partnerships. The case study is presented in two parts. The first part tries to explain, by means of the patterns, how the current governance and control mechanisms may have developed. In the second part, we study the control problems that arise from the introduction of a personal budget, and show how the Social Chart could provide a solution. We discuss three possible future scenarios for exploiting an information service like the Social Chart. The data for the case study was collected by semi-structured interviews with five experts from different health care organizations. The resulting $e^3$*-control* models were verified by the experts. In addition, we used publicly available policy documents, like [17], and government regulations.

### 3.1 Reverse Engineering the AWBZ

In the Netherlands, the AWBZ[4] deals with long-term and chronic diseases, such as protracted illness, invalidity, learning disability, mental disorders and geriatric diseases. Because this kind of care is too expensive to insure in a regular way, the system is arranged as public health care. A patient only pays a small part of the costs; the largest part of the costs is reimbursed to the care provider by a government fund, collected from taxes.

Suppose we have a hypothetical health care system (Figure 4). There are two parties: the patient and the care provider, for example a general practitioner. The patient receives care in return for fees. The corresponding process model is relatively simple: the patient pays only after receiving the care, according to the Execution Monitoring pattern, applied from the perspective of the patient (Table 3). So there is direct quality feedback. The task of paying the fees may be delegated to a trusted party, for example the family of the patient.

| Name | Execution Monitoring |
|---|---|
| Context | Patient (primary actor) receives care (counter object) from care provider (counter actor) in return for fees (primary object). |
| Problem | Patient is not certain if the care provider will provide care, as agreed. |
| Solution | Patient must verify that care is provided, before paying the fees. |

**Table 3.** Applying Execution Monitoring

Note that because of differences in expertise and status, the patient, or the family of the patient, are often not in a position to verify the quality of the care. Patients tend to trust care providers. Generally, care providers do not only provide care to get paid, but also to help patients. So, applying accounting models and professional distrust seems

[4] Algemene Wet Bijzondere Ziektekosten (Exceptional Medical Expenses Act)

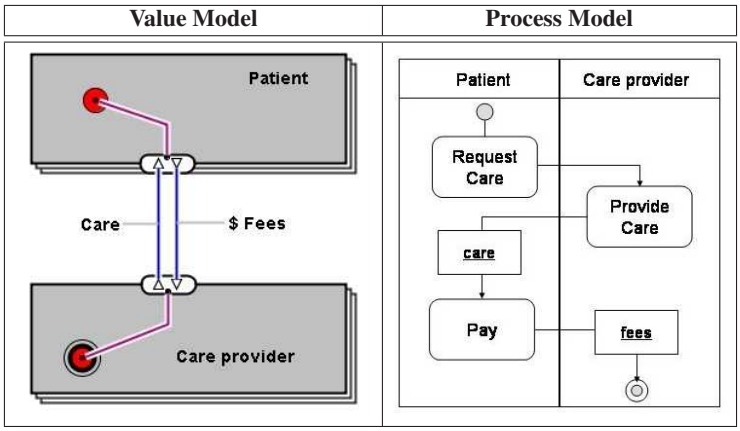



**Fig. 4.** Hypothetical Care Transaction

inappropriate for many care situations. Nevertheless, it may help to understand the governance and control of the health care system. This caveat applies throughout the rest of the case.

In this hypothetical situation, there is a problem. Not all citizens fall ill, but those who do, are faced with very high costs. For long-term and chronic diseases, these costs can not even be carried by individual health insurance policies. Some kind of solidarity is needed between healthy citizens and chronic patients, managed by the government [15]. Such an exceptional health care system is shown in Figure 5. The solidarity is shown by the fact that value transfers may be summed over a 'stack' of actors, for a specific time period.

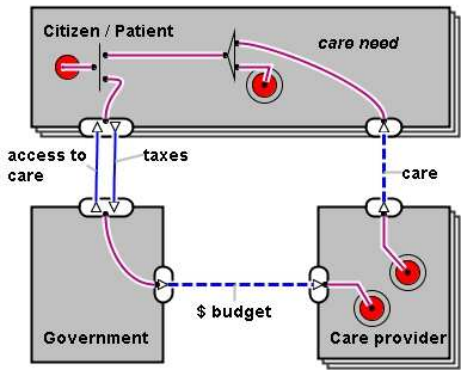



**Fig. 5.** Solidarity

There are essentially two value transfers. The first one, on the left, is concerned with access to health care. Citizens pay taxes, in return for the government funding care providers. When citizens do fall ill and become patients, depicted on the right, they can use the infrastructure. These possibilities are linked by a choice-fork (triangle). The model is sustainable, as long as the income from the taxes of citizens is sufficient to cover the average costs of providing care services ($\Sigma\ taxes \geq \Sigma\ budget$).

This is not a valid $e^3$-*value* model. First, the value transfer 'care', is a single value transfer. The reciprocal value transfer is made indirectly, through taxes. For this reason, there is no direct feedback about the quality of the care provided, to the actor who decides about funding. Such indirect dependencies are typical for the public sector. Second, there is no reciprocal evidence, that the care requested by a patient, is actually needed. In a regular insurance system, this would correspond to the verification that a claim is eligible according to the insurance policy and conditions. Third, for the value transfer 'budget' no reciprocal evidence is required to ensure accountability of the care provider. Moreover, in this model, funding of the care provider is not linked to the number of care services offered. Typically, a yearly budget is set.

We first focus on the control problems of the care provider.

*Control Problem 1a.* A care provider receives a budget, without being held accountable for the money. Control theory dictates, that every activity should be verified, with evidence of its execution [10][43]. Here, that means that the two dependency paths should be linked. On the basis of the control problem, that the care provider may receive a budget (primary object), that stands in no relation to the care actually provided (counter object), we therefore select the Execution Monitoring pattern (Table 4).

| Name | Execution Monitoring |
|---|---|
| Context | Government (primary actor) pays budget (primary object), so that the care provider (counter actor) may provide care to patients (counter object). |
| Problem | Government is not certain if the budget is in proportion to the care provided to patients. |
| Solution | Government must verify what care is provided, before paying the budget. The control activities 'witness' and 'verify' are added, as well as 'supporting document', namely the evidence. |

**Table 4.** Applying Execution Monitoring to Problem 1a

*Control Problem 1b.* All the costs of the care provider are reimbursed by the government. Therefore, there is no incentive for care providers to try and work more efficiently. Such an 'open-ended' system is one of the general reasons behind the increase in health care spending. Currently, there is much interest in budgeting schemes to reduce this problem, for example by output budgeting. We interpret budget agreements between government and care provider as a kind of mutual commitment, and select the Commitment Confirmation pattern (Table 5). Before committing to a certain budget, the government needs a commitment from the care provider that they can and will provide

care for such a budget. In some cases the budget is calculated by standardized Care Intensity Packages (ZZP).

| Name | Commitment Confirmation |
|---|---|
| Context | Government (primary actor) pays budget (primary object), so that the care provider (counter actor) may provide care to patients (counter object). |
| Problem | Care provider may claim to have no commitment to provide care for a given budget, and hence refuse to continue to provide care. |
| Solution | Government only commits to a specific budget, if care provider makes a commitment to provide the agreed care for that budget. Efficiency gains can be reinvested by the care provider. |

**Table 5.** Applying Commitment Confirmation to Problem 1b

The result of applying these two patterns is shown in Figure 6 (value network) and Figure 7 (process model). The task of controlling the budgets of care providers, and verifying the evidence, is delegated to an independent local agency, called Administration Office. For its administration task, the Administration Office receives a yearly budget from the government. This budget is fixed; it does not depend on the amount of care delivered by the care providers. By contrast, the budget of the care provider (1) depends on the number of care services actually delivered ($n$).

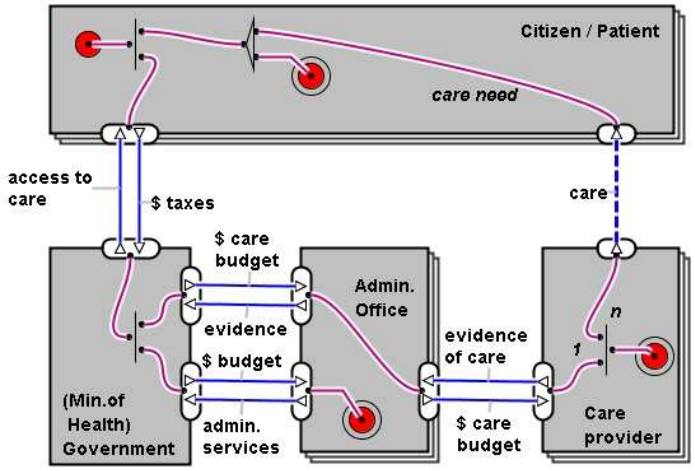



**Fig. 6.** Output Budgeting and evidence (value model)

*Control Problem 2a.* Now we concentrate on the patient. Although every citizen is entitled to health care when needed, access should be restricted to patients whose care is

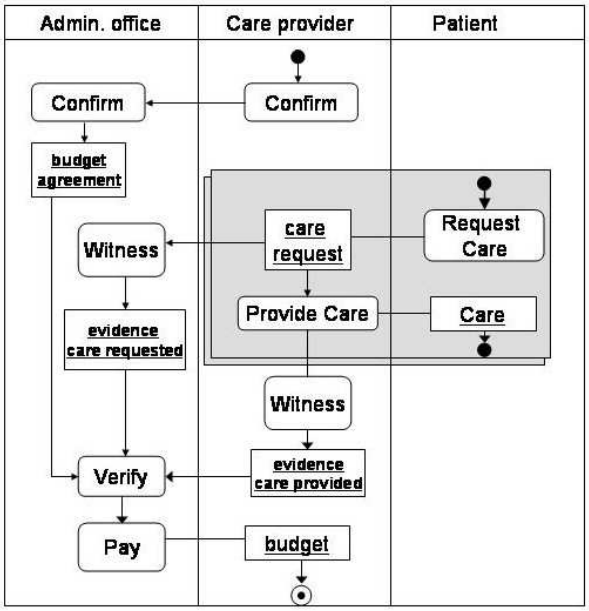



**Fig. 7.** Output Budgeting and evidence (process model)

(medically) necessary. Again we frame this problem as an instance of execution monitoring. The government or some government agency, must verify on the basis of evidence, like medical tests or a diagnosis from a physician, to which kinds of exceptional care the patient is entitled to.

*Control Problem 2b.* Initially, before March 2003, needs assessment was combined with the task of allocating actual care. Both were carried out by Regional Indication Centers (RIO). However, in some cases the needs assessments turned out to be inadequate: patients from different regions were given different products for the same diagnosis. These problems can be addressed by two applications of execution monitoring, combined with the general accounting principle of *standardization* [43]. In the AWBZ, care is categorized in standardized *functions*: domestic care, personal care, nursing, supportive assistance, activating assistance, treatment and institutional care. For each function, there are different *classes*, which specify the intensity of the treatment. Which care needs correspond to which functions and classes, is laid down in the Care Entitlement Regulation[5] .

In addition, a patient needs to pay a personal contribution. The solution, which more or less represents the current situation, is shown in Figure 8. Now there are two verification steps (Table 6): In the first step, Evidence of Care Needs (Supporting Document) is compared with the Care Entitlement Regulation (Document-to-be-verified). The result is modeled as a Right for Functions. This assessment is performed by a national

[5] Besluit Zorgaanspraken AWBZ

agency, called Central Indication Center (CIZ)[6]. In the second step, the local Administration Offices translates the Right for Functions into an actual Right for Care. Because the administration office has an overview of the available care providers in a region, this actor is in a position to advice patients on where to get the best care they need.

| Name | Execution Monitoring |
|---|---|
| Context | Government (primary actor) entitles all citizens (counter actor) to exceptional health care (primary object). |
| Problem | 2a. Government is not certain if the entitlement for care corresponds to actual care needs of a patient, as detailed in the Care Entitlement Regulation<br>2b. Needs assessment must be uniform, but the available care supply depends on the region. |
| Solution | Step 1. based on evidence of care need, CIZ makes an assessment of the care needs of a patient, and issues an evidence document: Right for Functions.<br>Step 2. The Administration Office translates the Right for Functions into Right for Care, allocating care services and care providers. |

**Table 6.** Applying Execution Monitoring to Problem 2a and 2b

*Control problem 3.* Until now, no actor in the network controls the quality of care products. As a result, provisioning of care of low quality could remain undetected. In the previous step, administrative evidence is used for budget control of care providers, but no evidence of the quality of care is used.

However, a basic form of quality control does exist. The government is in a position to select care providers. Before being allowed to enter the network in the first place, the ability of care providers to provide the care functions for which they are known, must be assessed. This accreditation is the result of applying the pattern Certification. Accreditation is delegated to the Health care Insurance Board (CVZ)[7]. So in Figure 8, the Administration Office can only assign patients to care providers who have an accreditation from the CVZ. The CVZ cannot provide real quality control, but it based on administrative evidence, it can at least ensure that the care provider has adequate facilities.

### 3.2 Personal Budget and the Social Chart

The exceptional health are system described above has a number of problems. The right for health are, through needs assessment, is disconnected from the care that is actually available. For each care provider, the budget from the government has a limit. Care providers do not have an incentive to provide services above their budget. Moreover, this supply-driven system results in a fragmented and unbalanced care supply in some regions. Therefore the government is moving towards a demand-driven system.

[6] Centrum Indicatiestelling Zorg (CIZ)
[7] College van zorgverzekeringen (CVZ).

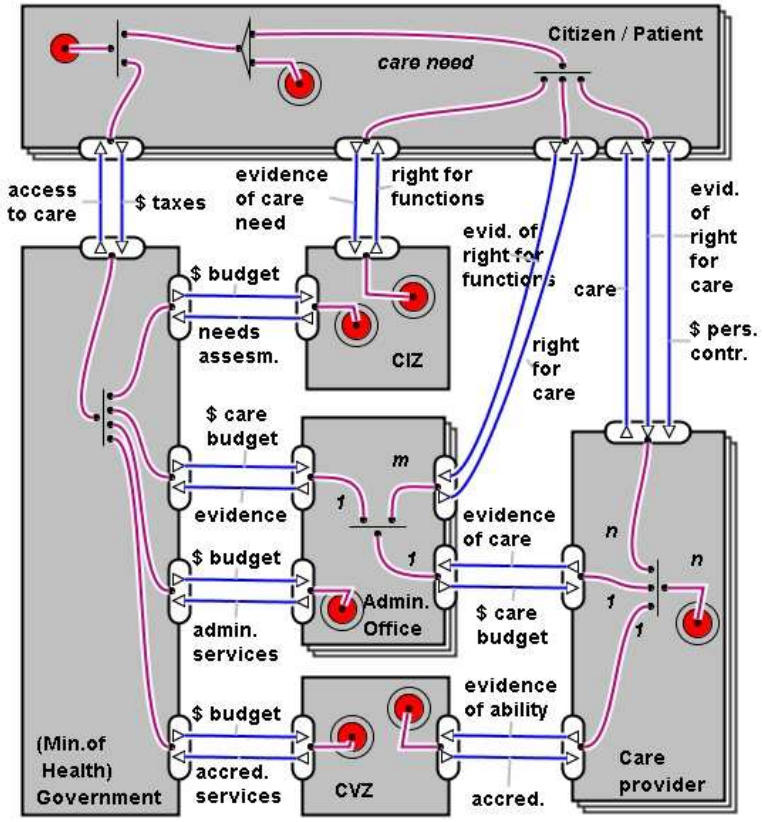



**Fig. 8.** Current Exceptional Health care Network

| Name | Certification |
|---|---|
| Context | Government (primary actor) provides budget (primary object) through Administration office, to care providers (counter actor), who in return provide care (counter activity) for patients. |
| Problem | Government is not certain if the quality of the care delivered corresponds to certain basic quality standards and requirements. |
| Solution | The basic abilities of the care provider are verified, The result is an accreditation. |

**Table 7.** Applying Certification to Problem 3

To try and solve these problems, the government introduced the possibility for patients to buy their own care services with a *Personal Budget*[8]. The personal budget is allocated by the Administration Office on the basis of the Right for Functions. The Personal Budget can cover care from any of the following six functions: domestic care, personal care, nursing, supportive assistance, activating assistance, and short term institutional care. The budget does not cover medical treatment, permanent institutional care or medication. Furthermore, the rules for care providers are liberalized. A care provider may now be any institution or private person. This liberalization has led to an enormous growth in the number of care providers, which creates more choice for patients, and in some cases allows them to regain control over their lives. We call these new care providers alternative care providers. This situation is shown in Figure 9.

From a control perspective, we can observe the quality control problem again. Because of the large number of alternative care providers, there is no way that the CVZ can accredit all of them. Therefore, alternative providers are generally not required to have an accreditation from the CVZ.

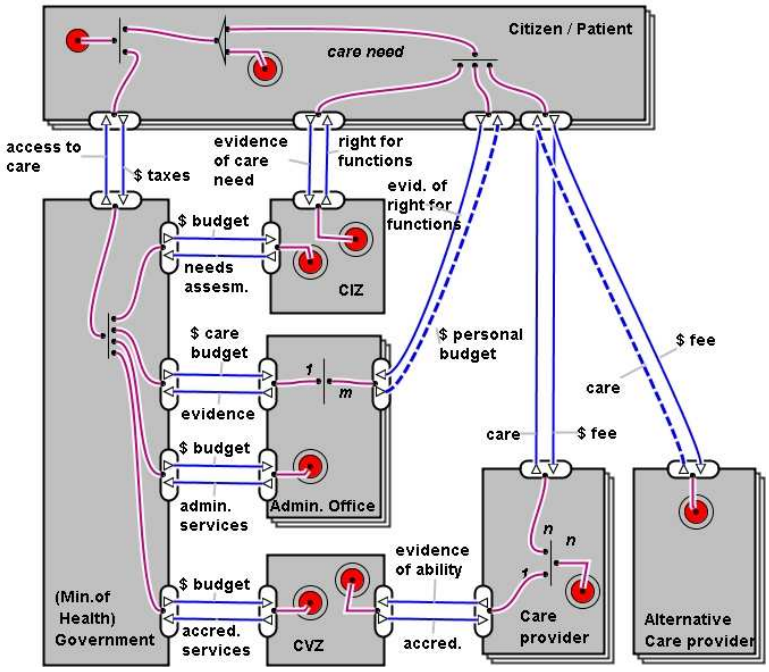



**Fig. 9.** Personal Budget

[8] Persoonsgebonden Budget (PGB)

*Control problem 4.* Patients and relatives are not adequately informed about the available care, and care providers in a region. Only information about accredited care providers is available from the Administration Office. Patients therefore tend to select traditional care providers, rather than alternative care providers. This may stifle the development of the market for alternative care providers. This observation corresponds to the general idea that *information asymmetry*, a situation in which the customer has less information about a product than the provider, has a negative effect on the emergence of new markets [3]. In Figure 9, this control problem is represented by a sub-ideal exchange (dashed line), labeled 'care'.

*Control problem 5.* There is a risk that low quality care is delivered by alternative care providers (dashed line for 'care'). We can also see the problem from the government's point of view. If patients select care providers that provide inadequate quality, public money is wasted. This is indicated by a dashed line for 'Personal Budget'.

In an e-commerce setting, the first problem would typically be solved by an information broker, who matches supply and demand. The second problem would be solved by an agency verifying reliability. Since these activities require special expertise, it makes sense to delegate them to a separate agent. To solve these problems, we must select a pattern from the pattern library. We dismissed the possibility of a regulatory body assessing quality, because of the large number of alternative care providers. So the Certification pattern can not be used. Control problem 4 (lack of information) seems to be related to the Partner Screening pattern. However, as it stands, the Partner Screening pattern does not deal with the general information needed to collect a set of feasible providers (see definition in Section 2.5). It only deals with the second half: reliability assessment. Now we could adapt the patterns to accommodate this problem, but in this validation test, we have chosen to keep the patterns as they are, and conclude that they do not completely cover all aspects of the case study.

Although not generated by a pattern, we do have a possible solution. The information problem can be solved by providing an information service, such as the Social Chart introduced at the beginning of this paper. In Figure 10 this solution is represented.

Regarding problem 5, quality control is a general concern in Dutch health care. Since 2004, an independent Health Inspectorate (not in the model) must supervise the quality of institutional care providers. But this organization cannot feasibly control the large number of alternative care providers. We therefore propose that an initiative like the Social Chart should enable a kind of informal quality control. It could provide, for example, a web-forum with testimonials, an online community peer review, a reputation mechanism, or collaborative filtering techniques [41]. In this manner, knowledge about the quality of care providers can be shared throughout the community of patients and relatives. Community-based quality control only works when users contribute to the community. That is why in the scenario shown in Figure 10, the Social Chart receives Quality Assessment from (some) patients.

### 3.3 Exploitation of the Social Chart

Figure 10 presents only one of many possible exploitation scenarios. The Social Chart could be set up for example by the patients' association, by commercial parties like an

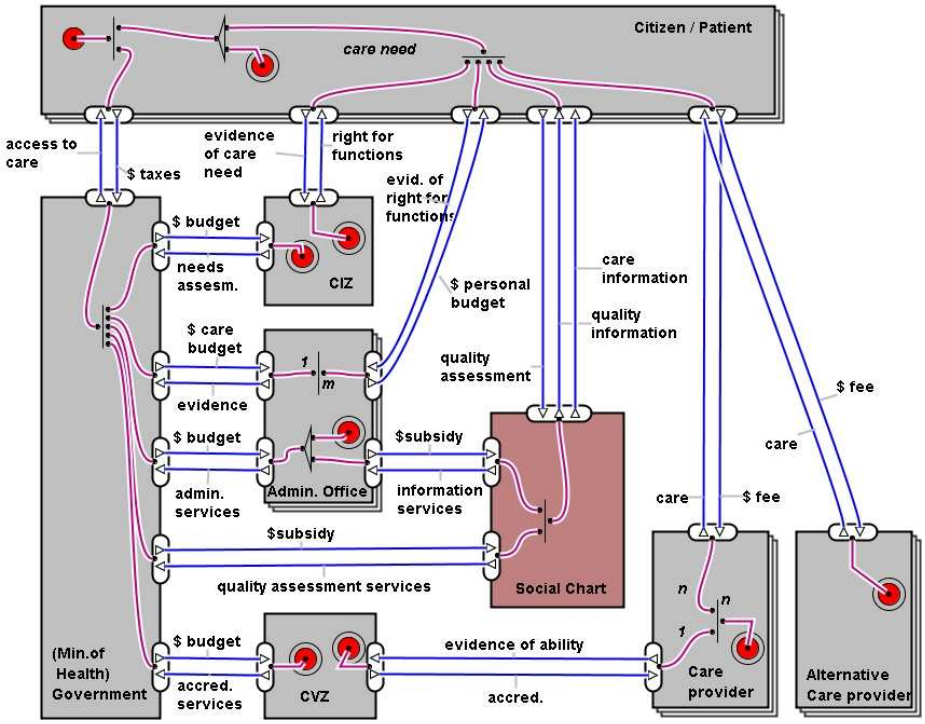



**Fig. 10.** Possible Scenario for Social Chart

insurance company or information broker, or by government. There are some indicators that virtual communities of patients are able to provide a form of community-based quality control [12]. There are already many successful virtual communities for patients with a chronic disease. For example, Leimeister et al [30] discuss the characteristics of a forum for cancer patients. Regarding dementia care, we find a relatively successful virtual community, hosted by patients association 'Alzheimer Nederland'. In the framework of the National Dementia Programme, Alzheimer Nederland collaborates with local government to improve dementia care [33]. So a combination of the patients' organization and local government seems a viable option for setting up and exploiting a kind of Social Chart.

Getting reliable online feedback is difficult [41]. Personal testimonials tend to be biased. Luckily, also quantitative approaches exist. For example, a Dutch information broker, independer.nl, is using a panel of general practitioners to get statistically valid feedback on the quality of hospitals.

### 3.4 Discussion

The objective of the case study is to validate the use of the control patterns. In particular, we are interested in modeling a highly regulated setting, with public-private partnerships.

*Regarding the use of Control patterns* Control patterns have a 'constructive' element, but reconstruction can also be used for analytic purposes. In this case study, we applied the control patterns to a hypothetical health care system based on solidarity. We have reengineered the administrative control mechanisms in the AWBZ. We have shown that crucial aspects of the AWBZ controls can be motivated by the application of control patterns. In particular, the evidence documents (needs assessment, evidence of ability to provide care) are generated by application of the Execution Monitoring pattern. Quality control is established by applying the Partner Screening pattern, or by applying the Certification pattern when executed by a regulatory body. Budget control arrangements can be seen as a kind of transaction, with mutual commitments laid down in a contract. In addition, the delegation of tasks to separate agencies makes a large difference. So although delegation is not a control pattern, it does play a role as an organizational pattern. However, providing information about care providers and the care supply, the first function of the Social Chart, could not be established as the result of a control pattern.

In addition, the analysis demonstrates that there is a quality 'control gap' for alternative care providers. We have indicated a particular solution to fill this control gap: an interactive Social Chart, which could provide information about care supply and care providers in region, and possibly provide community-based quality control. The use of quality feedback to make a decision can be interpreted as application of the Partner Screening pattern.

We have only highlighted some aspects, ignoring others. In particular, we have had little to say about the information aspects of the administrative procedures. For this case study, the control patterns are too much focused on processes and administrative control, and too little on information and evidence.

*Regarding highly regulated environments* The use of a value-based modeling technique in a highly regulated setting raises some issues for discussion. For more on these issues we refer to [26].

**Indirect reciprocity** In this case study, the economic reciprocal relation is often indirect. This is modeled in $e^3$-*value* by a broken dependency path (thin line). People pay only for access to a service (triangle). In any insurance, the insurance premiums (here from taxes) should cover the claims from patients.

**Regulatory rights** Public-private partnerships are heavily regulated. Regulation can take the form of a system of legal rights, to restrict access to a service. Examples in the case study are Right for Functions and Right for Care. Legal rights can be seen as a value object in the $e^3$-*value* ontology.

**Evidence documents** This case shows the need for various evidence documents. Although collection and interpretation of evidence is usually modeled as part of the regular business process, here evidence documents are seen as value objects.

**Community-based reciprocity** A community-based quality control, like a recommender system, only works when members contribute Sharing and exchanging information, like experiences about care providers, can be based on solidarity in a community.

**Control services** A control service like needs assessment can be seen as a separate service, which can be delegated. This is highlighted in the model. All parties, including government agencies, need to be funded

*Regarding Multi-agent Systems* Multi-agent systems are used when either the solution to an automation problem, or the problem itself is inherently distributed. In this paper we discuss a case study of the second kind. Although the AWBZ health care system used to be centralized, with a large influence for the government (Ministry of Health), the case study illustrates the development towards a more distributed structure. Semi-independent government agencies like the CIZ or the Administration Office, have taken over government tasks regarding access control the the AWBZ. In particular the introduction of the Personal Budget and the acceptance of alternative care providers show that the health care network is becoming more and more 'open'. For alternative care providers accreditation before being allowed to enter the network is no longer compulsory. On the other hand, quality control and information provision have become more important, now that access is no longer restricted to traditional care providers.

## 4 Conclusions

The health care sector is subject to a constant revision. In general, it is much harder to set up and maintain an information service, when the context is subject to change. When regulations change, the business opportunities may well change too. Therefore a generic method for analyzing and developing governance and control mechanisms for network organizations is needed. Control patterns provide such a method.

A *control pattern* is a description of a generic and re-usable control mechanism for a recurring control problem, to be selected on the basis of the context of application. Like

design patterns, control patterns capture 'best practices' in a domain. Based on accounting literature and various case studies, we have developed a representation language for expressing control patterns, and a library of generic control patterns.

In this paper we have validated the control patterns, on a case study in health care. We have reconstructed the development of the governance and control mechanisms of the AWBZ system for provision of exceptional care. The case study shows that crucial aspects of the administrative controls can be motivated by the control patterns. In particular, evidence documents, like the needs assessment are generated by the Execution Monitoring pattern. Quality control can be established by partner screening or certification. Budget control arrangements can be seen as the application of the Commitment Confirmation pattern, just like in business contexts. So the control patterns have proved to be useful and adequate in analyzing this case study.

However, there are also some limitations. The patterns focus on process aspects and administrative controls. Much less attention is paid to information and evidence collection. This is unfortunate for this case study, because information provision is one of the main functions of the Social Chart. Management information issues have not been studied in this paper.

*Acknowledgments* We would like to thank the members of the FrUX project for their valuable input, in particular Rose-Marie Dros and Franka Meiland of the Vrije Universiteit Medical Center (VUmc).

## References

1. Alexander, C. *The Timeless Way of Building*. Oxford, Oxford University Press, 1979.
2. Baida, Z. *Software-aided Service Bundling: Intelligent Methods and Tools for Graphical Service Modeling*. PhD thesis, Vrije Universiteit, Amsterdam, 2006.
3. Bakos, Y. The emerging role of electronic marketplaces on the internet. *Communications of the ACM*, 41(8):35–42, 1998.
4. Beedle, M. Pattern based reengineering. *Object Magazine*, January, 1997.
5. Boella, G., Hulstijn, J., Tan, Y.-H., and van der Torre, L. Transaction trust in normative multiagent systems. In Falcone, R. and Barber, S., editors, *Proceedings of the AAMAS workshop on Trust in Agent Societies (Trust'05), Utrecht.*, 2005.
6. Boella, G. and van der Torre, L. A game theoretic approach to contracts in multiagent systems. *IEEE Transactions on Systems, Man and CyBernetics - Part C*, 36(1):68–79, 2006. Special issue on Game-theoretic Analysis and Stochastic Simulation of Negotiation Agents.
7. Bons, R. W., Dignum, F., Lee, R. M., and Tan, Y.-H. A formal analysis of auditing principles for electronic trade procedures. *International Journal of Electronic Commerce*, 5(1):57–82, 2000.
8. Bons, R. W. H. *Designing Trustworthy Trade Procedures for open Electronic Commerce*. PhD thesis, University of Rotterdam, 1997.
9. Brckert, H.-J., Fischer, K., and Vierke, G. Holonic transport scheduling with TeleTruck. *Applied Artificial Intelligence*, 14(7):697725, 2000.
10. Chen, K. *Schematic Evaluation of Internal Accounting Control Systems*. PhD thesis, University of Texas at Austin, 1992. revised version available as Chen, K. and Lee, R.M. (1992), EURIDIS Research Monograph RM-1992-08-1.
11. Coplien, J. O. and Harrison, N. *Organizational Patterns of Agile Software Development*. Prentice Hall, 2004.

12. Dannecker, A. and Lechner, U. Success factors of communities of patients. In Ljungberg, J. and Andersson, M., editors, *Proceedings of the 14th European Conference on Information Systems (ECIS 2006)*, page 12 pages on CDROM. Göteborg University, Sweden, 2006.
13. Dastani, M., Hulstijn, J., Dignum, and F. Meyer, J.-J. C. Issues in multiagent system development. In *Proceedings of the Third International Joint Conference on Autonomous Agents and Multiagent Systems (AAMAS'04)*, pages 922 – 929. IEEE Computer Society, Washington, DC, 2004.
14. Dröes, R., Meiland, F., Doruff, C., Varodi, I., Akkermans, H., Baida, Z., Faber, E., Haaker, T., Moelaert, F., Kartseva, V., and Tan, Y.-H. A dynamic interactive social chart in dementia care. In Bos, L., Laxminarayan, S., and Marsh, A., editors, *Medical and Care Compunetics 2*, volume 114 of *Studies in Health Technology and Informatics*. IOS Press, 2005.
15. Drummond, M. *Methods for the economic evaluation of health care programmes*. Oxford University Press, 2005.
16. Eisenhardt, K. M. Agency theory: An assessment and review. *Academy of Management Review*, 14(1):57–74, 1989.
17. Exter, A., Hermans, H., and andR. Busse, M. D. Healthcare systems in transition: Netherlands. Technical report, WHO Regional Office for Europe, Copenhagen,, 2004.
18. Firozabadi, B. S. and van der Torre, L. Towards an analysis of control systems. In Prade, H., editor, *Proceedings of the Thirteenth European Conference on Artificial Intelligence (ECAI'98)*, pages 317–318, 1998.
19. Gambetta, D. *Trust*, chapter Can we trust trust?, pages 213–237. Basil Blackwell, New York, 1988.
20. Gamma, E., Helm, R., Johnson, R., and Vlissides, J. *Design Patterns: Elements of Reusable Object-Oriented Software*. Addison Wesley, Boston, 1995.
21. Gordijn, J. and Akkermans, J. Value-based requirements engineering: Exploring innovative e-commerce ideas. *Requirements Engineering*, 8(2):114–134, 2003.
22. Hu, X., Lin, Z., Whinston, A., and Zhang, H. Hope or hype: On the viability of escrow services as trusted third parties in online auction environments. *Information Systems Research*, 15(3):236–249, 2004.
23. Jones, A. and Carmo, J. Deontic logic and contrary-to-duties. In Gabbay, D., editor, *Handbook of Philosophical Logic*, pages 203–279. Kluwer, 2002.
24. Kartseva, V., Gordijn, J., and Tan, Y.-H. Towards a modelling tool for designing control mechanisms in network organisations. *International Journal of Electronic Commerce*, 10(2):57–84, 2005.
25. Kartseva, V., Gordijn, J., and Tan, Y.-H. Inter-organisational controls as value objects in network organisations. In *Proceedings of the 18th Conference on Advanced Information systems Engineering (CAiSE 2006), Luxembourg*, 2006.
26. Kartseva, V., Hulstijn, J., Gordijn, J., and Tan, Y.-H. Modelling value-based inter-organizational controls in healthcare regulations. In *Proceedings of the 6th IFIP conference on e-Commerce, e-Business, and e-Government, Turku, Finland, (I3E'06)*, 2006.
27. Kartseva, V., Hulstijn, J., Gordijn, J., and Tan, Y.-H. Towards value-based design patterns for inter-organizational control. In *Proceedings of the 19th Bled Conference: eValues (Bled'06)*, page CDROM, 2006.
28. Keil, P. Principal agent theory and its application to analyze outsourcing of software development. In *Proceedings of the International Workshop on Economics-Driven Software Engineering Research (EDSER'2005)*, pages 1 – 5. IEEE Computer Society, 2005.
29. Kolp, M., Giorgini, P., and Mylopoulos, J. Multi-agent architectures as organizational structures. *Journal of Autonomous Agents and Multi-Agent Systems*, 13(1):3–25, 2006.
30. Leimeister, J. M., Daum, M., and Krcmar., H. Towards mobile communities for cancer patients: the case of krebsgemeinschaft.de. *International Journal on Web Based Communities*, 1(1):58–70, 2004.

31. Malone, T. and Crowston, K. The interdisciplinary study of coordination. *ACM Computing Surveys*, 26(1), 1994.
32. Mayer, R., Davis, J., and Schoorman, F. An integrative model of organizational trust. *Academy of Management Review*, 20(3):709–734, 1995.
33. Meerveld, J., Schumacher, J., Krijger, E., Bal, R., and Nies, H. Werkboek landelijk dementieprogramma. Technical report, Nederlands Instituut voor Zorg en Welzijn (NIZW), 2004.
34. Meyer, J.-J. and Wieringa, R. *Deontic Logic in Computer Science: Normative System Specification*. John Wiley & Sons, 1993.
35. Moss, S., Gaylard, H., wallis, S., and Edmonds, B. SDML: A multi-agent language for organizational modelling. *Computational and mathematical Organization Theory*, 4(1):43–70, 1998.
36. Motschnig-Pitrik, R., Randa, P., and Vinek, G. Specifying and analysing static and dynamic patterns of administrative processes. In *Proceedings of the 10th European Conference on Information Systems (ECIS 2002), Gdansk, Poland*, 2002.
37. Osterwalder, A. *The Business Model Ontology: A Proposition in a Design Science Approach*. PhD thesis, University of Lausanne, Lausanne, Switzerland, 2004.
38. Parunak, H. V. D. A practitioners' review of industrial agent applications. *Autonomous Agents and Multi-Agent Systems*, 3(4):389–407, 2000.
39. Ronmey, M. and Steinbart, P. *Accounting Information Systems*. Prentice Hall, New Jersey, 10th edition, 2006.
40. Rumbaugh, J., Jacobson, I., and Booch, G. *The Unified Modelling Language Reference Manual*. Addison Wesley Longman, Reading, MA,, 1999.
41. Schubert, P. and Ginsburg, M. Virtual communities of transaction: Personalization in electronic commerce. *Electronic Markets*, 10(1):45–55, 2000.
42. Seruca, I. and Loucopoulos, P. Towards a systematic approach to the capture of patterns within a business domain. *Journal of Systems and Software*, 67:1–18, 2003.
43. Starreveld, R., de Mare, B., and Joels, E. *Bestuurlijke Informatieverzorging (in Dutch)*, volume 1. Samsom, Alphen aan den Rijn, 4th edition, 1994.
44. Tan, Y.-H. and Thoen, W. Formal aspects of a generic model of trust for electronic commerce. *Decision Support Systems*, 33(3):233 – 246, 2002.
45. Tapscott, D., Lowy, A., and Ticoll, D. *Harnessing the Power of Business Webs*. Harvard Business School Press, Boston, MA, 2000.
46. Weigand, H. and de Moor, A. Workflow analysis with communication norms. *Data and Knowledge Engineering*, 47(3):349–369, 2003.
47. Wooldridge, M. *An Introduction to Multiagent Systems*. John Wiley and Sons, Chichester, 2001.
48. Zambonelli, F., Jennings, N. R., and Wooldridge, M. Organisational rules as an abstraction for the analysis and design of multi-agent systems. *International Journal of Software Engineering and Knowledge Engineering*, 11(3):303–328, 2001.

## Appendix: Control Pattern Library

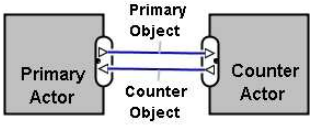
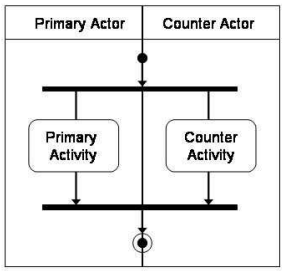
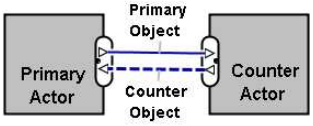
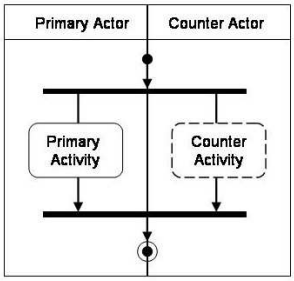
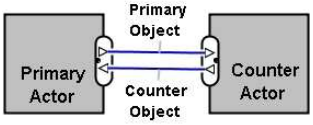
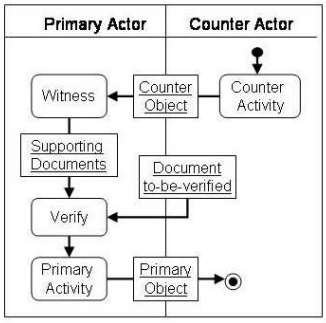
| Execution Monitoring | | |
|---|---|---|
| Context |  |  |
| Problem |  |  |
| Solution |  |  |

**Table 8.** Execution Monitoring

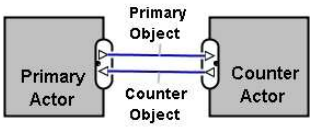
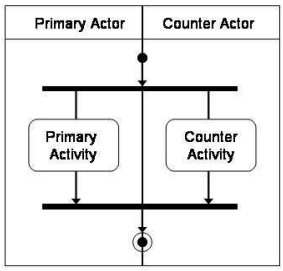
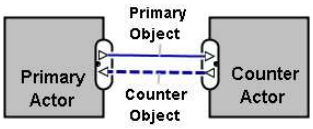
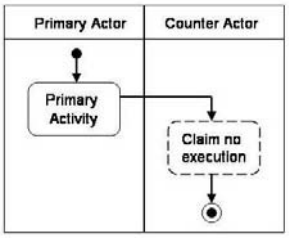
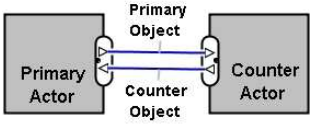
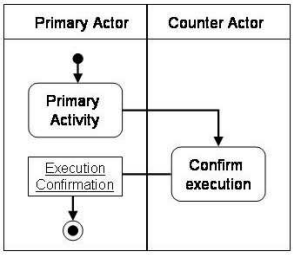
| Execution Confirmation | | |
|---|---|---|
| Context |  |  |
| Problem |  |  |
| Solution |  |  |

**Table 9.** Execution Confirmation

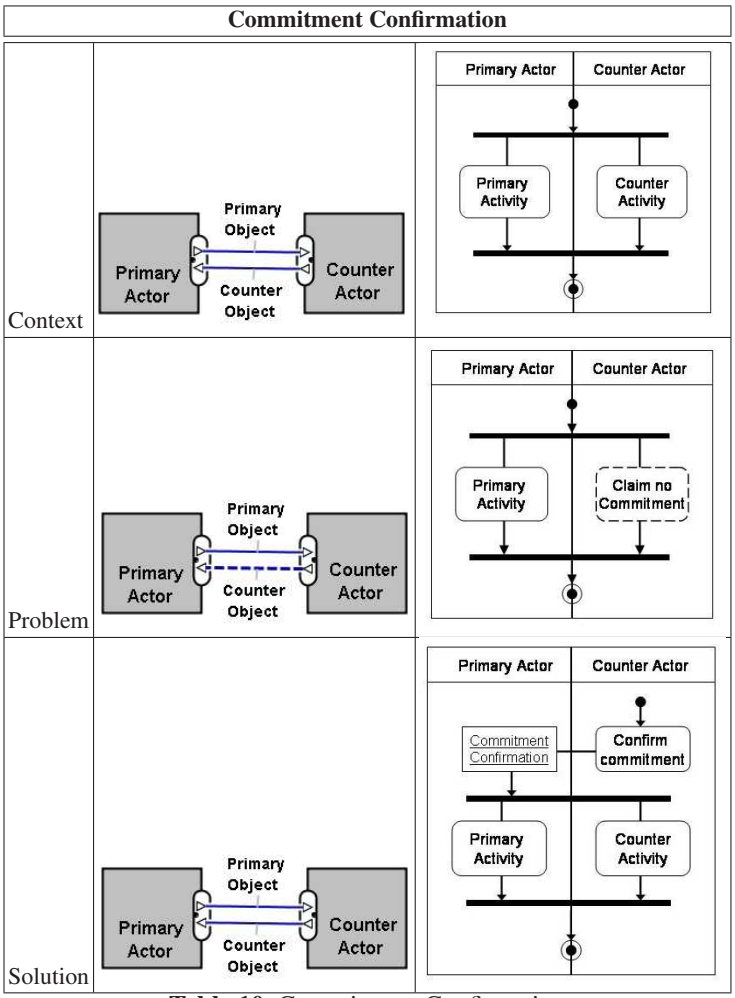



**Table 10.** Commitment Confirmation

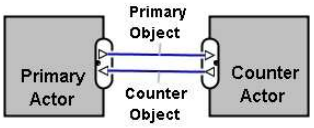
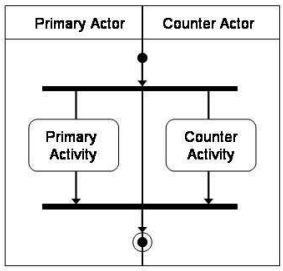
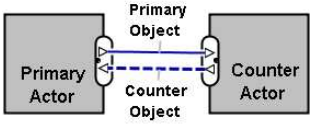
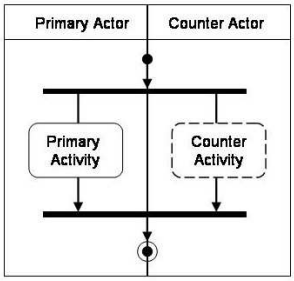
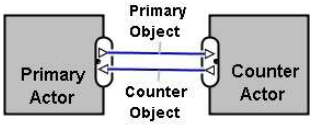
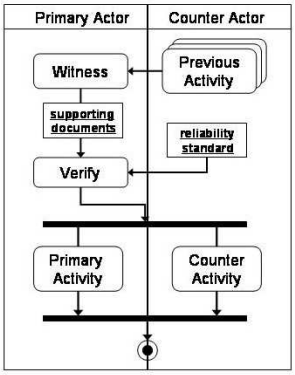
<table>
<tr><th colspan="3">Partner Screening</th></tr>
<tr><td>Context</td><td> 
</td><td> 
</td></tr>
<tr><td>Problem</td><td> 
</td><td> 
</td></tr>
<tr><td>Solution</td><td> 
</td><td> 
</td></tr>
</table>

**Table 11.** Partner Screening

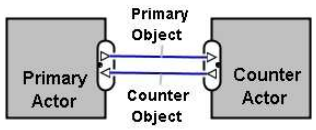
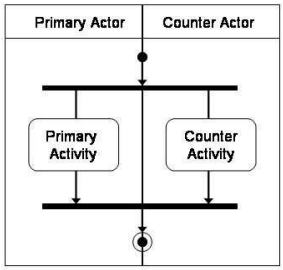
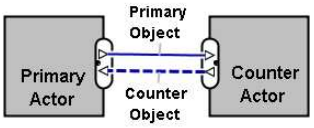
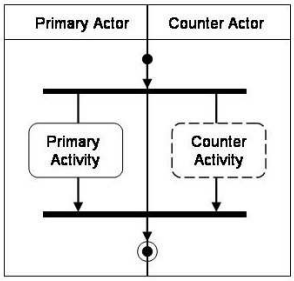
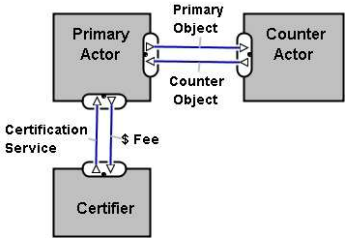
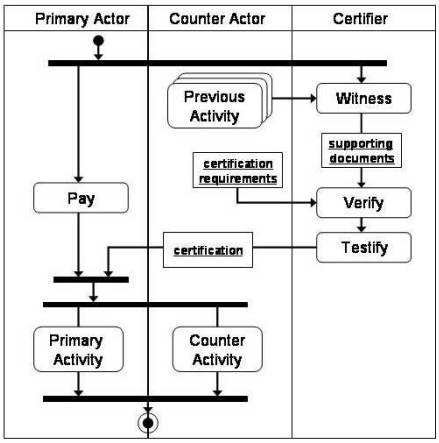
<table>
<tr><th colspan="3">Certification</th></tr>
<tr><td>Context</td><td> 
</td><td> 
</td></tr>
<tr><td>Problem</td><td> 
</td><td> 
</td></tr>
<tr><td>Solution</td><td> 
</td><td> 
</td></tr>
</table>

**Table 12.** Certification

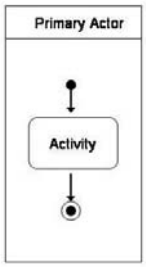
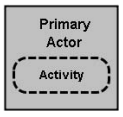
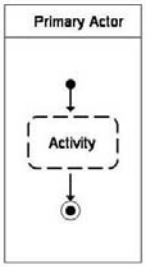
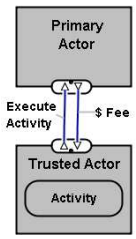
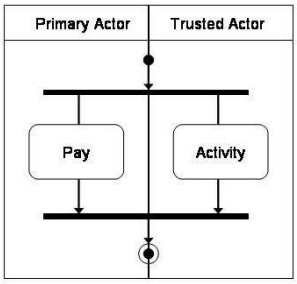
| Delegation (Organizational Design Pattern) | | |
|---|---|---|
| Context |  |  |
| Problem |  |  |
| Solution |  |  |

**Table 13.** Delegation (Organizational Design Pattern)

# Deriving individual obligations from collective obligations

Christophe Garion[1], Laurence Cholvy[2]

[1] SUPAERO
10 av. Édouard Belin
31055 Toulouse, France garion@supaero.fr
[2] ONERA-Toulouse
2 bis av. Édouard Belin
31055 Toulouse, France cholvy@cert.fr

**Abstract.** A collective obligation is an obligation directed to a group of agents so that the group, as a whole, is obliged to achieve a given task. The problem investigated here is the impact of collective obligations to individual obligations, i.e. obligations directed to single agents of the group. The groups we consider do not have any particular hierarchical structure nor have an institutionalized representative agent. In this case, we claim that the derivation of individual obligations from collective obligations depends on several parameters among which the ability of the agents (i.e. what they can do) and their own personal commitments (i.e. what they are determined to do). As for checking if these obligations are fulfilled or not, we need to know what are the actual actions performed by the agents.
This present paper addresses these questions in the rather general case when the collective obligations are conditional ones.

**Keywords.**

## 1 Introduction

This paper studies the relation between collective obligations directed to a group of agents and the individual obligations directed to the single agents of the group. We study this relation in the case when the group of agents is not structured by any hierarchical structure and has no representative agent like in [1].

According to Royakkers and Dignum [2], a collective obligation is an obligation directed to a group of individuals i.e. a group of agents. For instance (this is an example given by Royakkers and Dignum), when a mother says: "*Boys, you have to set the table*", she defines an obligation aimed at the group of her boys.

A collective obligation addressed to a group of agents is such that this group, as a whole, is obliged to achieve a given task. This comes to say that a given task is assigned as a goal to the group as a whole. In the mother's example, the goal assigned to the boys is to set the table and the mother expects that the table will be set by some actions performed by her boys. Whether only one of

Dagstuhl Seminar Proceedings 07122
Normative Multi-agent Systems
http://drops.dagstuhl.de/opus/volltexte/2007/911

her boys or all of them will bring it about that the table is set is not specified by the mother.

In particular, one must notice that in the example, the mother does not oblige each of her boys to set the table. This shows the difference between collective obligations and what Royakkers and Dignum call "restricted general obligations" which are addressed to every member of the group. For instance, "*Boys, you have to eat properly*" is not a collective obligation but a restricted general obligation directed to every mother's boy.

Norman and Reed [3] use the terms *collective group* and *distributive group* to make this dictinction. If distributive, a group is addressed distributively ("*Boys, you have to eat properly*"); if collective, a group is being addressed as a collective ("*Boys, you have to set the table*").

What is particularly interesting with collective obligations is to understand their impact on the individual obligations of the agents in the group, i.e. to understand when and how the collective obligations are translated into individual obligations. In the mother's example, will the eldest boy have to carry the forks and knifes, the second the glasses and the youngest the plates ? Or will the youngest have to carry everything ?

One can notice that when the mother directs the collective obligation to her boys, she does not direct (even implicitly) individual obligations to some or all of her boys. More generally, we think that when an agent directs a collective obligation to a group, it does not define individual obligations to some or all agents of the group. The consequence is that, in case of violation of the collective obligation, the only possible responsible towards the one who directed the obligation, is the group as a whole: no precise agent can be responsible of the violation of a collective obligation in front of the agent who directed that collective obligation.

However, we think that when the agents of a group with no hierarchical structure receive a collective obligation, they may coordinate themselves to provide a plan (or a task allocation), by committing themselves to make some actions. These commitments imply individual obligations that some agents must satisfy.

Understanding how the collective obligations are translated into individual obligations is the problem which is investigated here. We claim that the derivation of individual obligations from collective obligations depends on several parameters among which the ability of the agents (i.e. what each agent can do) and their own personal commitments (i.e. what each agent is determined to do). Latter on, by examining the actual actions of each agent of the group, one can check if these obligations are satisfied or violated.

For instance, if all the boys keep on watching TV (thus, do not set the table) then the collective obligation is violated. Notice that the collective obligation will be violated too if the eldest one, who is the tallest and the only one who can take the glasses, does not take the glasses, even if the two youngest boys carry the forks, the knifes and the plates. As said previously, in this case the whole group is responsible of the violation of the collective obligation. This can be questionable, particularly by the two youngest boys in the last case since they

will be all punished because of the eldest's actions. However we will show that, in this case, the eldest can be taken as responsible by the group because he was the only one able to take the glasses.

This present paper addresses the question of the translation of a collective obligation into individual obligations in the rather general case when the collective obligations are conditional ones. Roughly speaking, a conditional obligation is an obligation which applies when a given condition is true. For instance, "*If it is sunny, set the table in the garden; else set it in the dinner-room*" defines two conditional obligations: if it is sunny, the boys have to set the table in the garden but else, they have to set it in the dinner-room. In this work, we have chosen Boutilier's conditional preferences logic [4], [5], for representing conditional obligations.

So, this work assumes that a set of conditional obligations is directed to a group of agents. It also assumes a model of agents, describing each agent of that group by its knowledge about the current situation, its abilities and its commitments. It first defines a characterization of the obligations that the whole group have to satisfy. Then, grouping the agents according to their ability, it then defines the obligations that such sub-groups have to satisfy. Finally, given the commitments of the agents, it defines their individual obligations. As for checking if these obligations are satisfied or not, we have to consider the results of the agents' actions.

This work is based on the work of Boutilier [4], who addresses some of these questions in the case of a single agent. In that paper, Boutilier assumes a set of conditional preferences expressing a goal for a single agent. He then describes a way to define the actual goals of the agent, given what it knows (or more exactly, what it believes) and given what it controls. Like Boutilier mentions it, this work can be applied to deal with obligations instead of goals (cf. also [6]). Our aim is to adapt Boutilier's work in the case of collective obligations. This will lead us to enrich the model of agents by considering their commitments.

This paper is organized as follows. Section 2 quickly presents Boutilier's work and in particular $CO^*$ logic and the model of agent he considers. Section 3 adapts this work to the case of collective obligations and section 4 illustrates it on an example. Finally section 5 is devoted to a discussion.

## 2 A solution in the case of a single agent

This section quickly presents Boutilier's work in the case of a single agent. It first recall the semantics of the logic used by Boutilier then it recalls the model of agent he considers and its impact on the definition of goals.

### 2.1 *CO* and *CO** logics and conditional preferences

Given a propositional language $PROP$, Boutilier defines $CO$ logic whose language extends $PROP$ with two primitive modal operators: $\Box$ and $\overleftarrow{\Box}$. Models of

$CO$ are of the form: $M = \langle W, \leq, \phi \rangle$ where $W$ is a set of worlds, $\phi$ is a valuation function [1], and $\leq$ is a total pre-order[2] on worlds, allowing one to express preference: $v \leq w$ means that $v$ is at least as preferred as $w$.

**Definition 1.** *Let $M = \langle W, \leq, \phi \rangle$ be a CO model. The valuation of a formula in $M$ is given by:*

| | |
|---|---|
| $M \models_w \alpha$ | *iff* $w \in \phi(\alpha)$ *for any propositional letter* $\alpha$. |
| $M \models_w \neg\alpha$ | *iff* $M \not\models_w \alpha$ *for any formula* $\alpha$. |
| $M \models_w (\alpha_1 \wedge \alpha_2)$ | *iff* $M \models_w \alpha_1$ *and* $M \models_w \alpha_2$ *if* $\alpha_1$ *and* $\alpha_2$ *are formulas.* |
| $M \models_w \Box\alpha$ | *iff for any world* $v$ *such that* $v \leq w$, $M \models_v \alpha$ |
| $M \models_w \overleftarrow{\Box}\, \alpha$ | *iff for any world* $v$ *such that* $w < v$, $M \models_v \alpha$ |
| $M \models \alpha$ | *iff* $\forall w \in W \ M \models_w \alpha$. |

Thus, $\Box\alpha$ is true in the world $w$ iff $\alpha$ is true in all the worlds which are at least as preferred as $w$. $\overleftarrow{\Box}\, \alpha$ is true in the world $w$ iff $\alpha$ is true in all the worlds which are less preferred than $w$. Dual operators are defined as usual: $\Diamond\alpha \equiv_{def} \neg\Box\neg\alpha$ and $\overleftarrow{\Diamond}\ \alpha \equiv_{def} \neg\ \overleftarrow{\Box}\ \neg\alpha$. Furthermore, Boutilier defines: $\overleftrightarrow{\Box}\ \alpha \equiv_{def} \Box\alpha \wedge\ \overleftarrow{\Box}\ \alpha$ and $\overleftrightarrow{\Diamond}\, \alpha \equiv_{def} \Diamond\alpha \vee\ \overleftarrow{\Diamond}\, \alpha$.

Let $\Sigma$ be a set of formulas and $\alpha$ be a formula of $CO$. $\alpha$ is a logical consequence of (or deducible from) $\Sigma$ iff any model which satisfies $\Sigma$ also satisfies $\alpha$. It is denoted as usual: $\Sigma \models \alpha$.

Boutilier then considers $CO^*$ [5], a restriction of $CO$ by considering a class of $CO$ models in which any propositional valuation is associated with at least one possible world. The $CO^*$ models are $CO$ models $M$ which satisfy: $M \models\ \overleftrightarrow{\Diamond}\, A$, for any satisfiable formula $A$ of $PROP$.

In the following, we only consider $CO^*$.

In order to express conditional preferences, Boutilier considers a conditional connective $I(-|-)$, defined by:

$$I(B|A) \equiv_{def} \overleftrightarrow{\Box}\, \neg A \vee\ \overleftrightarrow{\Diamond}\, (A \wedge \Box(A \rightarrow B))$$

$I(B|A)$ means that if $A$ is true, then the agent ought to ensure that $B$.

An absolute preference is of the form: $I(A|\top)$[3]. It is denoted $I(A)$.

In order to determine its own goals, an agent must have a knowledge about the real world, or more exactly some beliefs about the real world. Boutilier thus introduces $KB$, a finite and consistent set of formulas of $PROP$, which expresses the believes the agent has about the real world. $KB$ is called a knowledge base.

Given $KB$ and given a model of $CO^*$, the most ideal situations are characterized by the most preferred worlds which satisfy $KB$. This is defined as follows:

[1] i.e. $\phi : PROP \rightarrow 2^W$ such that $\phi(\neg\varphi) = W - \phi(\varphi)$ and $\phi(\varphi_1 \wedge \varphi_2) = \phi(\varphi_1) \cap \phi(\varphi_2)$.

[2] $\leq$ is reflexive, transitive and connected binary relation.

[3] Where $\top$ is any propositional tautology.

**Definition 2.** *Let $\Sigma$ be a set of conditional preferences. Let $KB$ be a knowledge-base. An ideal goal derived from $\Sigma$ is a formula $\alpha$ of $PROP$ such that: $\Sigma \models I(\alpha|Cl(KB))$, where $Cl(KB) = \{\alpha \in PROP : KB \models \alpha\}$.*[4]

*Example 1.* Suppose an employee who is requested to make sure that the proposal for the financial management of the next year is written unless the statistics are not collated. We consider a propositional language whose letters are $s$ (the statistics for the current year are collated) et $fp$ (the proposal for the financial management of the next year is written) and we consider the two conditionals $I(fp)$ and $I(\neg fp|\neg s)$ which express that the proposal for the financial management for the next year should be written, but if the statistics of the current year are not collated, then it is preferred that it this proposal is not written. The possible worlds are: $w_1 = \{fp, s\}$, $w_2 = \{\neg fp, \neg s\}$, $w_3 = \{fp, \neg s\}$, $w_4 = \{\neg fp, s\}$[5]. Because of $I(fp)$, the worlds $w_1$ and $w_3$ can be the most preferred ones. But, due to $I(\neg fp|\neg s)$, $w_3$ cannot be one of the most preferred. Thus, $w_1$ is the only one most preferred, i.e. $w_1 \leq w_2$, $w_1 \leq w_3$ and $w_1 \leq w_4$. Furthermore, $w_3 \leq w_2$ is impossible because of $I(\neg fp|\neg s)$. Thus $w_2 \leq w_3$. The models which satisfy $I(fp)$ and $I(\neg fp|\neg s)$ are thus the following:

$$\begin{array}{ll} M_1: & w_1 \leq w_2 \leq w_3 \leq w_4 \\ M_2: & w_1 \leq w_2 \leq w_4 \leq w_3 \\ M_3: & w_1 \leq w_4 \leq w_2 \leq w_3 \end{array}$$

Assume first that $KB_1 = \{s\}$ (the statistics are collated). Thus $Cl(KB_1) = \{s\}$. Ideal goals for the agent are $\alpha$ such that $\forall M\ M \models I(\alpha|s)$. $fp$ is thus an ideal goal for the agent: since the statistics are collated, the agent has to write the financial proposal. Assume now that $KB_2 = \{\neg s\}$ (the statistics are not collated). One can prove that $\neg fp$ is now the ideal goal of the agent: since the statistics are not collated, the agent must not write the financial proposal. This is questionable and discussed in the following section.

Notice that this example is inspired from a scenario studied in [3]. Here, we modified this scenario in order to get conditional obligations and we restrict it to a single agent. Moreover, an extension of this scenario will be studied in section 4.

### 2.2 Controllable, influenceable propositions and CK-goals

By definition 2, any formula $\alpha$ such that $M \models I(\alpha|Cl(KB))$ is a goal for the agent. Boutilier notes that this is questionable if $KB$ is not "fixed" i.e. if the agent can change the truth value of some propositions in $KB$. For instance,

[4] In fact, Boutilier uses a non monotonic logic to deduce the default knowledge of the agent. Here, in order to focus to ideal goals, we restrict to classical logic.

[5] This way of denoting worlds is classical: for instance, $w_4 = \{\neg fp, s\}$ is a notation to represent $w_4 \notin \phi(fp)$ and $w_4 \in \phi(s)$.

in the second case of example 1, if the agent can collects statistics, it would be preferable that he does so, and that he also write the proposal in order to achieve the most preferred situation.

Boutilier then suggests, in the definition of $Cl(KB)$, to take into account only the propositions whose truth value cannot be changed by some of the agent's action.

Furthermore, it may happen that some formulas $\alpha$ which are characterized by definition 2 define situations that the agent cannot achieve. Assume for instance that in the first case of example 1, the agent cannot collect the statistics (for instance, s/he does have permission to access the database): collecting statistics cannot be a goal for the agent.

So, Boutilier introduces a partition of atoms of $PROP$ : $PROP = C \cup \overline{C}$. $C$ is the set of the atoms the agent controls (i.e. the atoms the agent can change the truth value) and $\overline{C}$ is the set of atoms the agent does not control (i.e. the atoms the agent cannot change the truth value)

For instance, if the agent has the access to the appropriate database and he can use the dedicated software, then we can consider that he controls the atom $s$. In any other case, we can consider that the agent does not control $s$.

**Definition 3.** *For any set of propositional letters $P$, let $V(P)$ be the set of all the valuations of $P$. If $v \in V(P)$ and $w \in V(Q)$ with $P$ and $Q$ two disjoint sets, then $v; w \in V(P \cup Q)$ is the valuation extended to $P \cup Q$.*

**Definition 4.** *Let $C$ and $\overline{C}$ respectively be the set of atoms that the agent controls and the set of atoms that he does not control. A proposition $\alpha$ is* controllable *iff, for any $u \in V(\overline{C})$, there are $v \in V(C)$ and $w \in V(C)$ such that $v; u \models \alpha$ and $w; u \models \neg\alpha$. A proposition $\alpha$ is* influenceable *iff there are $u \in V(\overline{C})$, $v \in V(C)$ and $w \in V(C)$ such that $v; u \models \alpha$ and $w; u \models \neg\alpha$.*

One can notice that for an atom, controllability and influenceability are equivalent notions. But this is not true for any non atomic propositions. Controllable proposition are influenceable, but the contrary is not true.

**Definition 5.** *The set of the uninfluenceable knowledge of the agent is denoted $UI(KB)$ and is defined by:*

$$UI(KB) = \{\alpha \in Cl(KB) : \alpha \textit{ is not influenceable}\}$$

In a first step, Boutilier assumes that $UI(KB)$ is a complete set, i.e. the truth value of any element in $UI(KB)$ is known.[6] Under this assumption, Boutilier then defines the notion of CK-goal:

**Definition 6.** *Let $\Sigma$ be a set of conditional preferences and $KB$ a knowledge base such that $UI(KB)$ is complete. A proposition $\varphi$ is a CK-goal (for the agent) iff $\Sigma \models I(\varphi|UI(KB))$ with $\varphi$ controllable (by the agent).*

[6] In a second step, Boutilier also examines the case when $UI(KB)$ is not complete. We will not focus on that case.

Finally, Boutilier notices that goals can only be affected by atomic actions, so it is important to characterize the set of actions which are guaranteed to achieve each CK-goal. So he introduces the following notion:

**Definition 7.** *An atomic goal set is a set $S$ of controllable atoms such that for any CK-goal $\varphi$, $\Sigma \models (UI(KB) \wedge S) \rightarrow \varphi$.*

*Example 2.* Consider again $\Sigma = \{I(fp), I(\neg fp|\neg s)\}$. Assume that $KB = \{\neg fp, \neg s\}$ (the statistics are not collected and the financial proposal is not written). Assume first that the agent can collect the statistics and also write the financial proposal. Then $UI(KB) = \emptyset$. Thus, $\{fp, s\}$ is the atomic goal set of the agent: the agent has to collect the statistics and to write the proposal. Assume now that the agent can collect the statistics but cannot write the financial proposal. Here, $fp$ is not controllable, thus $UI(KB) = \{\neg fp\}$ and the agent has no atomic goal.

## 3 Collective obligations

Let us now consider that the conditional preferences are modeling collective obligations which are allocated to a group of agents $\mathcal{A} = \{a_1, \ldots, a_n\}$. The problem we are facing now is to understand in which case these collective obligations define individual obligations and how to check if they are violated or not.

Following Boutilier, we will assume that each agent is associated with the atoms it controls and the atoms it does not control. But we extend that agency model by assuming that each agent is also associated with the atoms it commits itself to make true and the atoms it commits itself not to make true. These notions will be formalized latter, but intuitively, let us say that an agent commits itself to make an atom true if it expresses that it intends to perform an action that will make that atom true. An agent commits itself not to make an atom true if it expresses that it will perform no action that makes this atom true.

**Assumption 1** *In the following, the problem of determining individual obligations is studied assuming that the agents of the group have the same complete beliefs about the current world.*

### 3.1 Obligations of the group

Here, we extend the notion of CK-goals to the case when there are several agents. For doing so, we first extend the notions of controllability and influenceability to a group of agents.

Let $a_i$ be an agent of $\mathcal{A}$. Let $C_{a_i}$ be the set of atoms which are controllable by $a_i$ (i.e. atoms which $a_i$ can change the truth value) and $\overline{C}_{a_i}$ be the set of the atoms that are not controllable by $a_i$. The extension of notions of controllability and influenceability for a group of agent is given by the following definition.

**Definition 8.** *Let* $C = \bigcup_{a_i \in \mathcal{A}} C(a_i)$ *and* $\overline{C} = PROP \setminus C$. *A proposition* $\alpha$ *is controllable by the group* $\mathcal{A}$ *iff, for any* $u \in V(\overline{C})$, *there are* $v \in V(C)$ *and* $w \in V(C)$ *such that* $v; u \models \alpha$ *and* $w; u \models \neg\alpha$. *A proposition* $\alpha$ *is* influenceable *iff there are* $u \in V(\overline{C})$, $v \in V(C)$ *and* $w \in V(C)$ *such that* $v; u \models \alpha$ *and* $w; u \models \neg\alpha$.

This definition is obviously an extension of definition 4 to the multi-agent case.

*Example 3.* Consider a group of agents $\{a_1, a_2\}$ such that $p$ is controllable by $a_1$ and $r$ is controllable by $a_2$. We can show that the proposition $(p \vee q) \wedge (r \vee s)$ is not controllable by $\{a_1, a_2\}$. Indeed, if $q$ and $s$ are both true, whatever the actions of $a_1$ and $a_2$ are, the proposition will remain true. However, $p \wedge r$ and $p \vee r$ are both controllable by $\{a_1, a_2\}$.

Since we assume the the agents share a common belief about the current world, we can still consider a knowledge base $KB$ as a set of propositional formulas of $PROP$. And, like in the previous section, we assume that $KB$ is complete.

Like in [6], we can show that, given a Knowledge Base $KB$, some propositions are true and uninfluenceable in $KB$ even if they are influenceable according to definition 7.

*Example 4.* Consider a group $\{a_1, a_2\}$ such that $p$ is controllable by $a_1$ and $a_2$ and $q$ is not controllable neither by $a_1$ nor by $a_2$. According to definition 7, the proposition $p \vee \neg q$, even if not controllable, is influenceable by the group. Let us now consider $KB = \{p, \neg q\}$. $p \vee \neg q$ is true in $KB$ and, whatever the agents will do, will remain true. We will say that $p \vee \neg q$ is uninfluenceable in $KB$.

This leads to the extension of definition 7 as follows:

**Definition 9.** *Given a knowledge-base* $KB$, *a proposition* $\alpha$ *is influenceable in* $KB$ *iff there are* $u \in V(\overline{C})$ *such that* $u \models KB$, $v \in V(C)$ *and* $w \in V(C)$ *such that* $v; u \models \alpha$ *and* $w; u \models \neg\alpha$.

Notice that the previous example shows that a proposition, which is a logical consequence of $KB$ may be influenceable but uninfluenceable in $KB$.

We can thus introduce the following set:

**Definition 10.** *Let* $UI(KB)$ *be the set of logical consequences of* $KB$ *which are not influenceable by the group* $\mathcal{A}$ *or not influenceable in* $KB$ *by the group* $\mathcal{A}$.

**Definition 11. The group $\mathcal{A}$ has the obligation of $\varphi$ towards the agent who directed the collective obligation** *iff* $\Sigma \models I(\varphi | UI(KB))$ *with* $\varphi$ *controllable by* $\mathcal{A}$. *It is denoted* $O_{\mathcal{A}}\phi$.

This definition is obviously an extension of definition 6 to the multi-agent case. And we can check that it is also an extension, to the multi-agent case, of the notion of ideal obligations given in [6].

Thus, these obligations characterize the most preferred situation that the group $\mathcal{A}$ can achieve, given what is fixed and given what the whole group can control. But we can go further, by directing these obligations to the sub-groups that can really fulfill them.

**Definition 12.** *Let $\phi$ be a proposition. Let $\mathcal{A}_\phi$ be the union of the minimal subsets of $\mathcal{A}$ which controls $\phi$*[7]*. We say that* **the sub-group $\mathcal{A}_\phi$ has the obligation of $\phi$, towards $\mathcal{A}$** *iff $\Sigma \models I(\phi|UI(KB)$ . It is denoted $O^{\mathcal{A}}_{\mathcal{A}_\phi}\phi$.*

Thus, these obligations characterize the most preferred situation that the group $\mathcal{A}_\phi$ (a group which is the union of all the minimal sub-groups which control $\phi$) can achieve, given what is fixed.

*Example 5.* Let us now extend the example 1 to a group of agents and consider three agents *Alice, John* and *Tom* who are requested to write a financial proposal a scientifical proposal. The conditional preference which models this collective obligation is: $I(fp \wedge sf)$.

Assume that *Alice* is able to write the financial proposal while $John$ and $Tom$ are able to write the scientifical proposal. Then we have:

$$O_{\{Alice,John,Tom\}}(fp \wedge sp)$$
$$O^{\{Alice,John,Tom\}}_{\{Alice\}} fp$$
$$O^{\{Alice,John,Tom\}}_{\{John,Tom\}} sp$$

In other words, the group $\{Alice, John, Tom\}$ has the obligation to write the two proposals. The singleton $\{Alice\}$ has the obligation, towards the whole group to write the financial proposal, and the sub-group $\{John, Tom\}$ has the obligation, towards the whole group to write the scientifical proposal.

### 3.2 Agents commitments

Given an atom it controls, an agent may have three positions. The agent can express that it will perform an action making this atom true. We will say that the agent commits itself to make that atom true. The agent can also express that it will perform no action making this atom true. We will say that it commits itself not to make that atom true. Finally, it can happen that the agent does not express that it will perform an action making the atom true nor expresses that it will perform no action making it true. In this case, the agent does not commit itself to make the atom true, and does not commit itself not to make it true.

These three positions are modeled by three subsets of the sets of atoms that an agent controls. $Com_{+,a_i} \subseteq C_{a_i}$ is the set of atoms $a_i$ controls such that $a_i$

[7] We could also choose to define $\mathcal{A}_\phi$ as some of the minimal subsets of $\mathcal{A}$ which controls $\phi$. However, the whole study of the consequence of this alternative has not yet been done.

commits itself to make them true. $Com_{-,a_i} \subseteq C_{a_i}$ is the set of atoms $a_i$ controls such that $a_i$ commits itself not to make them true. $P_{a_i} = C_{a_i} \setminus (Com_{+,a_i} \cup Com_{-,a_i})$ is the set of atoms $a_i$ controls such that $a_i$ does not commit to make them true nor commits not to make them true.

These sets are supposed to be restricted by the following constraints:

**Constraint 1** $\forall a_i \in \mathcal{A}$ $Com_{+,a_i}$ *is consistent.*

**Constraint 2** $\forall a_i \in \mathcal{A}$ $Com_{+,a_i} \cap Com_{-,a_i} = \emptyset$

These two constraints are expressing a kind of consistency in the agent's model. By constraint 1, we assume that an agent does not commit itself to make something true and to make it false. By constraint 2, we assume that an agent does not commit itself to make an atom true and not to make it true.

*Remark 1.* The previous notions have been modeled in modal logic in [7], with two families of modal operators: $C_i$ and $E_i$, $i \in \{1 \dots n\}$. The operator $E_i$ is the *stit* operator ([8], [9]). $E_i\phi$ intends to express that the agent $a_i$ is seeing to it that $\phi$. It is defined by the following axiomatics:
(C) $E_i\phi \wedge E_i\psi \rightarrow E_i(\phi \wedge \psi)$ (T) $E_i\phi \rightarrow \phi$
(4) $E_i\phi \rightarrow E_iE_i\phi$ (RE) $\vdash (\phi \leftrightarrow \psi) \implies \vdash (E_i\phi \leftrightarrow E_i\psi)$
The operator $C_i$ is a KD-type operator and $C_i\phi$ intends to express that the agent $a_i$ commits itself to make $\phi$ true. It is defined by the following axiomatics:
(K) $C_i\phi \wedge C_i(\phi \rightarrow \psi) \rightarrow C_i\psi$ (D) $C_i\neg\phi \rightarrow \neg C_i\phi$
(Nec) $\vdash \phi \implies \vdash C_i\phi$

Given an atom $l$, and given these operators, an agent $a_i$ is facing three positions: $C_iE_il$, $C_i\neg E_il$ and $\neg C_iE_il \wedge \neg C_i\neg E_il$ (respectively, the agent commits itself to make $l$ true, i.e., the agent commits to make an action that makes $l$ true, the agent commits itself not to make $l$ true i.e. the agent commits itself to make no action that will make $l$ true, and the agent does not commit itself to make $l$ true nor commits itself not to make it true).

In this present paper, we forget this axiomatics and we only consider the three sets of atoms: $Com_{+,a_i}$, which corresponds to $\{l : C_iE_il\}$, $Com_{-,a_i}$, which corresponds to $\{l : C_i\neg E_il\}$, and $P_i$, which corresponds to $\{l : \neg C_iE_il \wedge \neg C_i\neg E_il\}$. But we can check that, by the previous axiomatics, we can derive, as a theorem, $\neg(C_iE_il \wedge C_iE_i\neg l)$. This explains constraint 1. We can also derive, as a theorem, $\neg(C_i\neg E_il \wedge C_iE_il)$. This explains constraint 2.

For defining individual obligations, we only need to consider the positive commitments (this assumption will be discussed in section 5). So, let us define:

**Definition 13.**

$$Com_{+,\mathcal{A}} = \bigcup_{a_i \in \mathcal{A}} Com_{+,a_i}$$

By this definition, $Com_{+,\mathcal{A}}$ is composed by any atom an agent commits itself to make true.

**Assumption 2** *In the following, $Com_{+,\mathcal{A}}$ is assumed to be consistent.*

This constraint is imposed in order to avoid the case when one agent commits itself to make an atom $a$ true, while another agent commits itself to make that atom false.

### 3.3 Individual obligations

We can now characterize the obligations that are directed to some agents of the group, given the obligations of the group and given the agent's commitments. Individual obligations are defined by:

**Definition 14.** *Let $\phi$ be a proposition such that $O_{\mathcal{A}}\phi$ holds. Let $a_i$ be an agent of $\mathcal{A}$. If there is some minimal $\{l_1, \ldots, l_m\} \subseteq Com_{+,a_i}$ such that $\models l_1 \wedge \ldots \wedge l_m \rightarrow \phi$, we say that $a_i$* **is obligated to satisfy** *$l_1 \wedge \ldots \wedge l_m$* **towards** *$\mathcal{A}_\phi$. This is denoted by $O_{a_i}^{\mathcal{A}_\phi}(l_1 \wedge \ldots \wedge l_m)$.*

Several remarks can be done on this definition. Let us suppose that the whole group $\mathcal{A}$ has the obligation to make $\phi$ true, (thus the sub group $\mathcal{A}_\phi$ has the obligation towards $\mathcal{A}$ to make $\phi$ true). Let us suppose that there is an agent $a_i$ such that there is some minimal $\{l_1, \ldots, l_m\} \subseteq Com_{+,a_i}$ such that $\models l_1 \wedge \ldots \wedge l_m \rightarrow \phi$.

First, the set $\{l_1, \ldots, l_m\}$ is said to be minimal, in the sense that for any $j \in \{1, \ldots, m\}$ $(\{l_1, \ldots, l_m\} - \{l_j\}) \not\models \phi$. $\{l_1, \ldots, l_m\}$ can be viewed as a kind of "prime implicant" of $\phi$, because we do not want to derive individual obligations with no link with the obligation imposed to the group. For instance, we can have $O_{\mathcal{A}}(fp)$: the group have the obligation to do a financial proposal. Let us suppose that $Tom$ commits itself to do the financial proposal and to wash its car. In this case, with no minimality condition, we could derive that $Tom$ is obligated to do the financial proposal and to wash its car towards the sub-group of agents "reponsible" for doing the financial proposal. But the washing of Tom's car has no link with the collective obligation of doing a financial proposal. The minimality condition ensures that the individual obligations will not be "out of context".

If an agent $a_i$ commits itself to achieve some "actions" $l_1, \ldots, l_m$ such that $\models l_1 \wedge \ldots \wedge l_m \rightarrow \phi$, then it has the individual obligation towards $\mathcal{A}_\phi$ to make $l_1 \wedge \ldots \wedge l_m$ true (notice that in [10], the agent has the individual obligation to make $\phi$ true. The new formulation of individual obligation is more precise). This intuitively represents the fact that, since the sub-group $\mathcal{A}_\phi$ has the obligation to make $\phi$ true and since $a_i$ commits itself towards the other members of $\mathcal{A}_\phi$ to make some sufficient conditions of $\phi$ true, then it has now the obligation, towards $\mathcal{A}_\phi$ to make those sufficient conditions true.

Finally, let us remark that such an agent $a_i$ belongs to $\mathcal{A}_\phi$, because it controls some literals (in fact $l_1, \ldots, l_m$) whose conjunction implies $\phi$.

### 3.4 Satisfactions and violations

For checking if the different obligations introduced previously are violated or not, we must examine the results of the agents' actions.

Let $KB_{next}$ be the state of the world resulting from the actions of the agents. Let $\phi$ such that $O_{\mathcal{A}}\phi$.

- if $KB_{next} \models \phi$ then the collective obligation is not violated. We say that the collective obligation is fulfilled.
- if $KB_{next} \not\models \phi$ then $O_{\mathcal{A}}(\phi)$ is violated.
  The whole group $\mathcal{A}$ is taken as responsible of the violation, by the agent who directed the collective obligation.
  We consider $\mathcal{A}_\phi$. Since we have $O_{\mathcal{A}}\phi$ we also have $O^{\mathcal{A}}_{\mathcal{A}_\phi}(\phi)$. Thus, since $KB_{next} \not\models \phi$, this proves that $O^{\mathcal{A}}_{\mathcal{A}_\phi}(\phi)$ is violated too. And $\mathcal{A}_\phi$ is taken as responsible, by $\mathcal{A}$, of this violation.
- let us consider all the agents $a_i$ such that there is some $\varphi$ such that $O^{\mathcal{A}_\phi}_{a_i}(\varphi)$. If $KB_{next} \not\models \varphi$, the obligation $O^{\mathcal{A}_\phi}_{a_i}(\varphi)$ is violated too and $a_i$ can be taken as responsible by $\mathcal{A}_\phi$ of the violation of its commitment i.e. $O^{\mathcal{A}_\phi}_{a_i}\varphi$.
  Moreover, if $KB_{next} \not\models \phi$, $a_i$ can be taken by $\mathcal{A}_\phi$ of the violation of $O^{\mathcal{A}}_{\mathcal{A}_\phi}(\phi)$.

## 4 Study of an example

In this section, we will illustrate the previous definitions by an example. Let us consider a group $\mathcal{A}$ of three agents named Alice (denoted by $A$), John (denoted by $J$) and Tom (denoted by $T$). That group is addressed the following obligations:

- if the statistics are collected, then the financial proposal and the scientifical proposal should be written.
- if the statistics are not collected, then the financial proposal should not be written, but the scientifical proposal should be.

Let us denote by $s$ the fact "the statistics are collected", by $fp$ the fact "the financial proposal is written" and by $sp$ the fact "the scientific proposal is written". The previous scenario is translated into the following set of $CO^*$ formulas: $\{I(fp \wedge sp|s), I(\neg fp \wedge sp|\neg s)\}$.

Let us examine some scenarios:

1. let us suppose that $KB = \{s, \neg fp, \neg sp\}$. The statistics are collected, but neither the financial proposal nor the scientific proposal are written. Let us also suppose that $C_A = C_T = \{fp\}$ (i.e. Alice and Tom can write the financial proposal) and that $C_J = \{sp\}$ (i.e. only John can write the scientific proposal). So $\mathcal{A}$ controls both $fp$ and $sp$.
   In this case, $UI(KB) = \{s\}$ and $\mathcal{A}$ has the obligation of $fp \wedge sp$, thus $\mathcal{A}$ has the obligation of $fp$ and the obligation of $sp$. Moreover, as $fp$ is controllable by both Alice and Tom, then $\{A, T\}$ has the obligation towards $\mathcal{A}$ to achieve $fp$. Finally, as John is the only agent which controls $sp$, $\{J\}$ has the obligation towards $\mathcal{A}$ to achieve $sp$.
   Thus the obligations are : $O_{\mathcal{A}}(fp \wedge sp)$, $O^{\mathcal{A}}_{\{A,T\}}(fp)$ and $O^{\mathcal{A}}_{\{J\}}(sp)$.

(a) let us suppose that the agents do not commit themselves to anything. Let us also suppose that Alice, John and Tom do nothing. In this case, $KB_{next} = KB = \{s, \neg fp, \neg sp\}$.
As $fp$ and $sp$ are parts of the obligation $O_{\mathcal{A}}(fp \wedge sp)$, the collective obligation is then violated. $\mathcal{A}$ is taken as responsible of this violation. Moreover, as $\{A, T\}$ should have written the financial proposal $(O^{\mathcal{A}}_{\{A,T\}}(fp))$, $\{A, T\}$ is taken as responsible by $\mathcal{A}$ of the violation of $O_{\mathcal{A}}(fp)$. By the same way, $\{J\}$ is taken as responsible of the violation of $O_{\mathcal{A}}(sp)$ by $\mathcal{A}$.

(b) let us suppose that the agents do not commit themselves to anything. Let us also suppose that Alice writes the financial proposal and that John and Tom do nothing.
In this case, $KB_{next} = \{s, fp, \neg sp\}$ and $KB_{next} \models fp \wedge \neg sp$. The collective obligation imposed on $\mathcal{A}$ is violated and the group is taken as responsible of the violation of $fp \wedge sp$.
More precisely, $O_{\mathcal{A}}(fp)$ is fulfilled because Alice wrote the financial proposal. But $O_{\mathcal{A}}(sp)$ is violated. As previously, $\{J\}$ is taken as responsible of the violation of $O_{\mathcal{A}}(sp)$ by $\mathcal{A}$.

(c) let us suppose that Alice commits herself to write the financial proposal. In this case, $Com_{+,A} = \{fp\}$ and we can derive $O^{\{A,T\}}_{A}(fp)$ (because $O_{\mathcal{A}}(fp)$ holds). Alice is obligated to achieve $fp$ towards $\{A, T\}$.
Assume that Alice writes the financial proposal, that John writes the scientific proposal and that Tom does nothing. In this case, $KB_{next} = \{s, sp, fp\}$ and all the obligations are fulfilled.
Assume now that Alice does not write the financial proposal, but that Tom writes the financial proposal. Assume also that John writes the scientific proposal. In this case, the collective obligation $O_{\mathcal{A}}(fp \wedge sp)$ is satisfied, $O^{\mathcal{A}}_{\{A,T\}}(fp)$ is satisfied too, but $O^{\{A,T\}}_{A}(fp)$ is violated. Even if the group fulfilled its obligations, the obligation of Alice towards $\{A, T\}$ to achieve $fp$ is violated.
Let us finally suppose that Alice, John and Tom do nothing. In this case, as $KB_{next} = \{s, \neg sp, \neg fp\}$, the collective obligation for the group is violated. John has also violated its obligation toward the group $\mathcal{A}$ to do $sp$. Finally, $\{A, T\}$ has violated its obligation to do $fp$ toward $\mathcal{A}$ and Alice has violated its obligation to do $fp$ toward $\{A, T\}$.

2. let us now suppose that $KB = \{\neg s, \neg fp, \neg sp\}$, i.e. the statistics are not collected and neither the financial proposal nor the scientific proposal are written. Let us also suppose that $C_A = \{fp\}$, $C_J = \{sp\}$ and $C_T = \{s\}$. Thus $\mathcal{A}$ controls $l$, $p$ and $s$.
As $UI(KB) = \phi$, there are three obligations for $\mathcal{A}$ : $O_{\mathcal{A}}(s \rightarrow fp)$, $O_{\mathcal{A}}(\neg s \rightarrow \neg fp)$ and $O_{\mathcal{A}}(sp)$. Thus we have : $O^{\mathcal{A}}_{\{J\}}(sp)$, $O^{\mathcal{A}}_{\{A,T\}}(s \rightarrow fp)$ and $O^{\mathcal{A}}_{\{A,T\}}(\neg s \rightarrow \neg fp)$.
Let us suppose that Tom commits himself to collect the statistics. As $\models s \rightarrow (\neg s \rightarrow \neg fp)$, then Tom has the obligation towards $\{A, T\}$ to do $s$, i.e., $O^{\{A,T\}}_{\{T\}} s$.

Suppose now that Alice wrote the financial proposal (because she thought that Tom would collect the statistics) but that Tom does not collect them. Suppose also that John does nothing. Then, $KB_{next} = \{\neg s, fp, \neg sp\}$. In this case, $O_{\mathcal{A}}(\neg s \rightarrow \neg fp)$ and $O_{\mathcal{A}}(sp)$ are both violated by $\mathcal{A}$. John violated also his obligation $O^{\mathcal{A}}_{\{J\}} sp$ and $\{A, T\}$ violated its obligation $O^{\mathcal{A}}_{\{A,T\}}(\neg s \rightarrow \neg fp)$. But Tom can be taken as responsible by $\{A, T\}$ (and in particular by Alice) of the violation of $O^{\mathcal{A}}_{\{A,T\}}(\neg s \rightarrow \neg fp)$, because $O^{\{A,T\}}_{\{T\}} s$ holds.

## 5 Discussion

In this paper, we have presented a preliminary work about collective obligations, i.e. obligations directed to a group of agents.

We have assumed that there was no hierarchical structure in the group, and no institutionalized agent who represents the group like in [1]: the group is made of real agents who may coordinate or not to act on the world.

In this work, the collective obligations are represented by conditional preferences. The first step was to determine the obligations of the group, given what is fixed in the world and given what this group as a whole, can do. Then we considered that, if the group is obliged to make $A$ true, then it induces another obligation to the very sub-group who control $A$: that sub-group is obliged, towards the whole group, to make $A$ true. These definitions of obligation are direct extensions, to the multi-agent case, of one definition provided by Boutilier in the single-agent case.

As for individual obligations, they are induced as soon as an agent commits itself to satisfy, by one of its action, an obligation of the group. Checking if these obligations are violated or not need to consider the state of the world obtained after the agents' actual actions.

This work could be extended in many directions.

For instance, concerning the agent's model, it would be interesting to relate the notion of commitment used here with the notion of proposition which are "controllable and fixed" defined in [6]. We could also refine the notion of commitment to the notion of commitment toward a group of agent. For instance, Tom can commit himself to write the financial proposal toward Alice and John et commit himself to wash the car toward his wife. Using this distinction could refine the obligations derivation process. Moreover, we only consider that the agents commit themselves to make a literal true. We could extend this to propositional formulas. But in this case, the derivation process is more complicated. For instance, if an agent commits him/herself to do $a \vee b$ and $a$ and $b$ are implicants for two different obligations, it is difficult to determine what the other agents should do.

Secondly, one must notice that the notion of controllability taken here has an important weakness: if $l$ is controllable, then $\neg l$ is also controllable. This is questionable since having the ability to make an atom true does not necessarily mean having the ability to make its negation true. For instance, even if one can send emails, he/she cannot send emails back once they are sent.

We are currently working on a more refined model of ability in which an agent may control an atom but not its negation. In this refined model, we also intend to take into account the fact that some atoms are controllable not by a single agents but by a coalition of agents [11]: for instance, several agents (in several areas) are needed to collect statistics. The impact of this refinement to the previous work remains to be studied.

Notice also that the agents share the same beliefs about the world. What happens when the group of agents does not share a common set of beliefs ? For instance, Alice may believe that the statistics are not collected and act in this way and Tom may believe that the statistics are collected and write the financial proposal. To solve such conflicts, we could for instance use some kind of merging methods which are used to build a common belief set from several belief sets which can be contradictory [12,13,14]. Particularly, we suppose that the agents beliefs are knowledge, in the sense that what they believe is true in the real world. We could also suppose that the agents beliefs do not fit the actual world. In this case, is an agent responsible of some obligations violations if it has acted as its beliefs were true?

Concerning the definition of individual obligations, we only use the "positive" commitments of the agents. But each agent can also express commitment of the kind "I commit myself not to do the financial proposal". As there are two sets $Com_+$ and $Com_-$ for each agent, there must be two kinds of individual obligations. In our formalism, we express individual obligations to *do* something but no individual obligations *not to do* something.

## References


1. Carmo, J., Pacheco, O.: Deontic and action logics for organized collective agency, modeled through institutionalized agents and roles. Fundamenta Informaticae **48** (2001) 129–163
2. Royakkers, L., Dignum, F.: No organization without obligations: how to formalize collective obligation? In Klusch, M., Kerschberg, L., eds.: 11th International COnference on Databases and Expert Systems Applications (LNCS-1873), Springer-Verlag (2000) 191–207
3. Norman, T., Reed, C.: Group delegation and responsability. In: Proceedings of the first International Joint Conference on Autonomous Agents and Multiagent Systems (AAMAS'02), ACM Press (2002) 491–498
4. Boutilier, C.: Toward a logic for qualitative decision theory. In Doyle, J., Sandewall, E., Torasso, P., eds.: Principles of Knowledge Representation and Reasoning (KR'94), Morgan Kaufmann (1994) 75–86
5. Boutilier, C.: Conditional logics of normality : a modal approach. Artificial Intelligence **68** (1994) 87–154
6. Cholvy, L., Garion, C.: An attempt to adapt a logic of conditional preferences for reasoning with contrary-to-duties. Fundamenta Informaticae **48** (2001) 183–204
7. Cholvy, L., Garion, C.: Strategies for distributing goals in a team of cooperative agents. In Gleizes, M.P., Omicini, A., Zambonelli, F., eds.: Proceedings of the Fifth International Workshop on Engineering Societies in the Agents World (ESAW'04). Number 3451 in Lecture Notes in Artificial Intelligence, Springer-Verlag (2005) 178–190

8. Belnap, N., Perloff, M.: Seeing to it that: a canonical form for agentives. Theoria **54** (1988) 175–199
9. Horty, J., Belnap, N.: The deliberative stit : a study of action, omission, ability and obligation. Journal of Philosophical Logic **24** (1995) 583–644 Reprinted in *The Philosopher's Annual, Volume 18-1995*, Ridgeview Publishing Company, 1997.
10. Cholvy, L., Garion, C.: Collective obligations, commitments and individual obligations: a preliminary study. In Horty, J., Jones, A., eds.: Proceedings of the $6^{th}$ International Workshop on Deontic Logic In Computer Science ($\Delta$EON'02), Londres (2002) 55–71
11. Kraus, S., Shehory, O.: Methods for task allocation via agent coalition formation. Artificial Intelligence **101** (1998) 165–200
12. Konieczny, S., Pino-Pérez, R.: Merging information under constraints: a qualitative framework. Journal of Logic and Computation **12** (2002) 773–808
13. Cholvy, L., Garion, C.: Answering queries addressed to merged databases: a query evaluator which implements a majority approach. In Hacid, M.S., Raś, Z., Zighed, D., Kodratoff, Y., eds.: Foundations of Intelligent Systems - Proceedings of the $13^{th}$ International Symposium on Methodologies for Intelligent Systems, ISMIS 2002. Volume 2366 of Lecture Notes in Artificial Intelligence., Springer (2002) 131–139
14. Cholvy, L., Garion, C.: Querying several conflicting databases. Journal of Applied Non-Classical Logics **3** (2004) 295–327

# Designing Organizations: Towards a Model

**Draft Paper**

Emanuele Bottazzi[1,2], Roberta Ferrario[1], Claudio Masolo[1] and Robert Trypuz[1,3]

[1] Institute for Cognitive Sciences and Technologies of the National Research Council, Laboratory for Applied Ontology
38100, Trento, Via Solteri, 38, Italy
[2] University of Torino, Philosophy Department
10124, Torino, via S. Ottavio, 20, Italy
[3] University of Trento, Department of Information and Communication Technology
38050, Povo - Trento, Via Sommarive, 14, Italy
bottazzi;ferrario;masolo;trypuz@loa-cnr.it

**Abstract.** The purpose of this paper is to draw a preliminary model of an ontology of organizations. The emphasis is on the structural aspects of organizations and the relations that these have with the design process of the organization itself on the one hand, and with its normative layer on the other.

**Keywords.** Ontology, organizations, structure, design, norms

## 1 Introduction

This paper tries to lay the basis for an abstract model that integrates and accommodates different features of organizations which are only separately considered in the literature. In particular, we focus on a specific kind of organizations, namely social entities that are *designed* to obtain certain *objectives* coordinating some collective behavior by means of *norms*. The FIAT company or Al Quaeda are examples of the kind of organizations we intend to capture. For the time being, we don't consider 'emergent' organizations or self-organized groups (e.g. a group of friends meeting every Thursday at Mollie's pub) even though, probably, most of the aspects considered in our model are also relevant for this kind of organizations.

In our model, an organization is intended as:

- *multi-layered*: structured in irreducible layers (not reducible to basic roles and their interrelations);
- *designed*: created by means of a decisional process and with specific objectives in mind;
- *agentive*: coordinating agents in order to obtain its objectives;
- *realized*: ultimately built by autonomous agents playing specific roles;
- *situated*: immersed in the environment;

Dagstuhl Seminar Proceedings 07122
Normative Multi-agent Systems
http://drops.dagstuhl.de/opus/volltexte/2007/904

- *dynamic*: its structure and its realization may change through time;
- *regulated*: governed and structured by norms.

In this paper we focus on how the structure and the design of organizations are linked to the normative dimension. For this goal, other features need to be considered, but here we don't describe them in detail.

## 2 Organizations as Multilayered Entities

In literature, it is common to consider organizations as structured entities. Sociology [1], philosophy [2,3,4] and computer science[5,6,7,8] generally agree that organizations are (at least) complex sets of roles tied together. For example, the organization model of OperA [7] consists of a *social structure*, i.e. roles and groups of roles, and an *interaction structure*, which contains the interaction relations between the elements of the social structure. Similarly, in [9, p. 146] we read that "an organization is structured through a set of roles, to which are associated deontic notions (...), that apply to the agents that are the actual holder of such roles, when playing those roles". However, their structures are, in some sense, *flat*. This does not mean that these authors disregard the fact that roles can be arranged in a hierarchy (this is often the case in their models, e.g. in OperA [7] roles are hierarchically arranged by dependencies or power relations holding among them), but, at the end, organizations can be reduced to sets of interrelated roles. This is clearly true also in [10] where a function which defines the goals of the MultiAgent System as a whole starting from the goals of its agentive constituents is introduced.

Even though in our model roles are still the basic units of organizations, *sub-organizations* (i.e. organizations that are at an abstraction level which is in the middle between the one of the whole organization and the one of roles) play a fundamental role in the specification of organizations. In particular, in the model teleological considerations drive the structuring of organizations. Roles and sub-organizations are created and designed with the precise aim of accomplishing certain objectives whose achievement brings about the achievement of the overall objectives of the organization[1]. But, in real cases, the chosen structure and the chosen reallocation of objectives to sub-organizations do not offer a complete description of the whole organization. The goals of the whole organization represent the *common* goals of the sub-organizations and they cannot be reduced to a composition of the particular goals of the sub-organizations. This means that the whole organization is something more than its parts. In this sense, organizations are not only structured but also *multi-layered* entities and roles can be seen just as *unstructured* organizations.[2]

[1] These intuitions are already present in classical works as [11], more recently in [12] and [13].

[2] The presence of these overall objectives (pre-established and not emergent) could represent a motivation for distinguishing organizations from simple groups of roles.

The literature does not consider the latter features in an integrated fashion. In [6] the authors propose an organization ontology for the TOVE enterprise model in which organizations are seen as composite entities whose parts are divisions and subdivisions; on the other hand, the teleological aspects are explained with reference to goals of the organization and subgoals which are assigned directly to roles, but roles are not linked to sub-divisions, thus leaving the teleological and the structural dimensions independent . Similarly, in Enterprise Ontology (EO) [14] an organization – called organization unit – can be decomposed into smaller organizations (persons can be seen as organization units), so that an organization structure is built by means of management links[3] between organization units, but the decomposition of goals – called in EO 'purposes' – is not taken into account.

Furthermore, we also represent the context, the environment, in which the organization is *situated*. Already in [15] it is stressed that what they call the "conceptualized environment" must be part of the organizational model. In addition, some works in sociology, as [16,17], observe that the behavior of an organization is not only determined by its internal structure, but also by the way in which it is linked to other organizations that are not under its (complete) control.

As roles and sub-organizations can be linked together by relations like *dependence*, *trust*, *delegation*, etc., in our model we allow also the same kinds of relations with external organizations. These external links represent the environment. The arguments of these relations can be organizations, objectives, states of affairs, actions, etc. As an example, an organization $o_1$ could delegate just a specific objective to $o_2$, or $o_1$ could trust $o_2$ for performing specific (kinds of) actions, etc.

## 3 Organizations as Designed Entities

Organizations can also be seen as *artifacts* whose function is to coordinate some collective behavior. If we take an artifact, for instance a chair, we can see that each part of the chair contributes to the main function of the chair, that is something to sit on. In the same way, we can imagine that every part of the organization has a function that contributes to the general goal of the organization. For the time being we take this only as a metaphor, leaving aside some more detailed ontological questions on artifacts, namely whether organizations are *really* artifacts and to what extent they are different from material artifacts as chairs and hammers[4].

In this section we focus only on the *design process*: a designer starts by figuring out an organization with some general objectives and successively refines that organization by introducing new sub-organizations (with new objectives) linked in a specific (and maybe normative) way. In order to model design, [15]

[3] "To manage" here means "to assign purpose to".

[4] It is also possible to draw a parallel with the algebraic specification and program development, see for example [18], [19].

introduces a notion of *refinement* and characterizes it as a process of specification of the description of an organization. Organizations that (unstructured) are roles at a specific level of refinement can be structured and detailed at a lower level of refinement. In this sense, the design and the structure of an organization are linked, i.e. the designer expresses a specific way of 'implementing' the objectives of the whole organization by imposing a structure on it. He establishes how the objectives declared for the whole organization can be 'decomposed' into simpler objectives attributed to simpler sub-organizations and how these sub-organizations are linked by means of institutional relations. In general, this can be a much more complex relation than the standard *and/or* decomposition considered in most of the existing approaches. But a more precise link can be established between the refinement and the layers of organizations. Each refinement step can be seen as a change of layer in the organization structure, i.e. at each refinement step we are trying to implement the organization by introducing a new set of roles linked in a specific way, and we want to maintain this information in the resulting organization. This is done by decoupling the goals of the starting organization from the goals of the sub-organizations introduced in the refinement. In this sense the refinement can be seen as a link between two *flat* organizations. Clearly the designer wants to implement organizations in a correct way, i.e. he wants the refined organization to be able to achieve the goals declared for the initial one. The irreducibility of the goals of the whole organization to a composition of the goals of its sub-organizations offers all the necessary information to check the 'correctness' of the implementation, once we have a sort of composition function of goals.

Let us consider a simple example. The design process can be seen as a sequence of refinement steps: $o \rightsquigarrow o'$ stands for "$o'$ is the refinement of $o$". If $o$ is already structured (even though flat), we write $o \overset{o_1}{\rightsquigarrow} o'$, in order to indicate that this refinement is relative only to the $o_1$ component of the structured organization $o$.

Table 1 illustrates two different design processes of the organization $o$. At $t_1$, the first step of the design (a.) introduces two sub-organizations ($o_1$ and $r_3$) that are not linked, i.e. $o$ is refined in $o^1$, a flat organization exactly structured in $o_1$ and $r_3$. At $t_2$, the second step of the design (a.) refines only one component of $o^1$, namely $o_1$, introducing two non linked sub-organizations ($r_2$ and $r_3$). The multilayered organization that is obtained via this design process is depicted in figure 1.a. Similarly, the design process in table 1.b originates the multilayered organization in figure 1.b. If we reduce organizations to roles (unstructured organizations), then the two organizations in figure 1 are identical, but by considering them as multilayered we can also encapsulate the refinement in the structure. This is especially important in the case of organizations that are created by different designers by means of specific laws.

So far we considered only the refinement of goals. Other kinds of refinements can also be introduced: refinement of norms, refinement of the environment, etc. In addition, it is possible to consider other design operators like, for example, 'generalizing', 'deleting', etc. In particular, here as in [15] the process of speci-

fication of an organization is strictly top-down. But we can also consider some bottom-up operators, as for example the 'grouping' operator in [20], i.e. two or more organizations can be grouped together to achieve some common goals. We recognize the existence and importance of both processes: in fact a designer may decompose the overall objective of an organization into sub-objectives that he assigns to sub-organizations or roles that are purportedly created to accomplish those objectives. Otherwise, agents that have compatible, complementary or coinciding objectives can decide to share their objectives and join in a plural entity. In our model this difference is encoded only in the design representation, the obtained organization does not depend on the direction of the design. For example, table 2 illustrates the bottom-up design of the same organizations depicted in the figure 1[5]. Our multilayered model is then compatible with both kinds of design and therefore with the two approaches in the theory of organization singled out by [21]: organizations from aggregations of agents vs. organizations as designed entities that influence the behavior of agents.

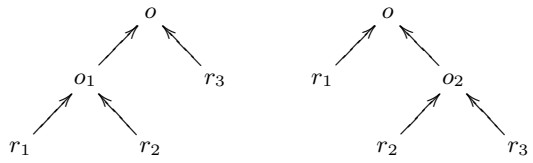
| **Time** | **Org.Intr.** | **Refin.** |
|---|---|---|
| $t_0$ | $o$ | |
| $t_1$ | $\{o_1, r_3\}$ | $o \overset{o}{\rightsquigarrow} o^1$ |
| $t_2$ | $\{r_1, r_2\}$ | $o^1 \overset{o_1}{\rightsquigarrow} o^3$ |

**a.**

| **Time** | **Org.Intr.** | **Refin.** |
|---|---|---|
| $t_0$ | $o$ | |
| $t_1$ | $\{r_1, o_2\}$ | $o \overset{o}{\rightsquigarrow} o^2$ |
| $t_2$ | $\{r_2, r_3\}$ | $o^2 \overset{o_2}{\rightsquigarrow} o^4$ |

**b.**

**Table 1.** Different refinement processes.




**a.** **b.**

**Fig. 1.** Different organizations resulting from the design processes in table 1.

[5] The grouping operator is represented by $\rightarrowtail$.

| Time | Org.Intr. | Group. |
|---|---|---|
| $t_0$ | $\{r_1, r_2\}$ | |
| $t_1$ | $o_1$ | $r_1, r_2 \rightarrowtail o_1$ |
| $t_2$ | $\{o, r_3\}$ | $o_1, r_3 \rightarrowtail o$ |

**a.**

| Time | Org.Intr. | Group. |
|---|---|---|
| $t_0$ | $\{r_2, r_3\}$ | |
| $t_1$ | $o_2$ | $r_2, r_3 \rightarrowtail o_2$ |
| $t_2$ | $\{o, r_1\}$ | $o_2, r_1 \rightarrowtail o$ |

**b.**

**Table 2.** Different grouping processes.

## 4 Organizations and Norms

In some accounts, as shown in [4], organizations are intended as completely made up of norms. For the moment we don't commit to such a strong position, but in any case undoubtedly norms are central in organizations and there are several ways in which the normative layer affects the organization and the behavior of its members. Here we just sketch some preliminary analyses of the following relevant topics.

*Norms and design.* Often the entire design of an organization is described by norms, as for example in many legal organizations. This kind of norms can be called, following Searle's terminology [22,23], "constitutive". Constitutive norms (or rules, as he calls them) are norms that, as pointed out in [24] and in [25], create new objects [25]:

> they have a defining function: they create new concepts, roles, social individuals; they can also establish which are the requirements that an entity should meet in order to be classified under a certain role or concept.

They can even create new organizations as, for example, the Republic of Italy and its Constitution[6].

*Norms and the structure of organizations.* By means of norms the objectives of the organizations are linked to their roles. The relations among these objectives and the roles can be permissions or obligations: as an example, a president is allowed to enter in some area but it is mandatory for the president not to interfere with the job of the technicians in that area. They can be understood as what Searle [22,23] calls "regulative rules" and Hart calls "primary rules" [30]. This kind of rules regulate antecedently existing forms of behavior. For instance, a rule like "drive on the right hand side of the road" regulates the driving without defining it: the driving exists before the rule that imposes duties to individuals by the way of the role that these individuals play.

[6] Formal frameworks for constitutive rules have been proposed, for instance in [26], [27], [28] and [29].

*Norms and contracts.* Agents' behavior is not only forced by regulative or primary norms. In what we can call 'realizations' of organizations, agents are often linked to the organizations via agreements or contracts. A realization is then a particular instance of a designed organization in which all the roles are assigned to specific agents. For instance, all the persons that actually have a (direct or indirect) employment contract with FIAT are its actual realization. In this sense, a contract can be conceived of as the bridge between the descriptive level of designed organizations and the concrete level of agents, i.e. a sort of norm that links abstract roles with specific individual agents. When an agent, say Paolo Rossi, is hired by a company, say FIAT, he acquires new rights and obligations that are partially specified in the contract he signs.

*Norms and inter-organizational relations.* As we stated in section 2, organizations are situated in an environment of other organizations that can be internal or external with respect to it. When two organizations are linked, this often affects the normative dimension. This kind of relations are considered in [25] and in [8]. In [25] an example of a specific iterorganizational normative relation is given, namely the relation that holds between the Italian State and the University of Torino [25]:

> We could say that the University of Torino is in a way "nested" into the Italian State. The normativity of the relation relies on the fact that the descriptive system of the "contained" organization is, in some sense, more specialized with respect to the descriptive system of the "containing" one: all the norms that are valid in the Italian State must also be valid in the University of Torino.

There are other kinds of norms and normative relations in the institutional setting, for example we can consider again contracts and agreements that make alliances possible, as in the case of military alliances among countries, like in the NATO organization.

*Metanorms and organizational change.* An important notion for understanding the complex relationships among organizations and norms is that of metanorm or what Hart calls "secondary rule". Secondary rules are, according to Hart [31], rules about rules. This kind of norms state – for example – how to resolve controversies in conflicting norms. At a certain time, after some steps of refinement, the designer may decide which is the structure that the organization will have. But organizations in a changing environment must be flexible. Nonetheless, not all the changes should be admissible. In this case it is possible to introduce some meta-norms that regulate the evolution of an organization by describing its acceptable changes. In addition to that, the designer itself can be subject to some norms that constraint the design itself.

## References

1. Scott, W.R.: Institutions and Organizations. Sage, Thousand Oaks, CA (2001)

2. Ladd, J.: Morality and the ideal of rationality in formal organizations. The Monist **54** (1970) 488–516
3. Tuomela, R.: The Philosophy of Social Practices. Cambridge University Press, Cambridge, UK (2002)
4. Miller, S.: Social institutions. In Zalta, E.N., ed.: The Stanford Encyclopedia of Philosophy. (2007)
5. van der Torre, L., Hulstijn, J., Dastani, M., Broersen, J.: Specifying multiagent organizations. In: Proceedings of the Seventh International Workshop on Deontic Logic in Computer Science (DEON'04), to appear. (2004)
6. Fox, M.S., Barbuceanu, M., Gruninger, M., Lin, J.: An organisation ontology for enterprise modelling. In Carley, K., Gasser, L., eds.: Simulating Organizations: Computational Models of Institutions and Groups. AAAI/MIT Press, Menlo Park, CA (1998) 131–152
7. Dignum, V.: A Model for Organizational Interaction: based on Agents, founded in Logic. PhD thesis, Universiteit Utrecht (2004)
8. Boella, G., van der Torre, L.: Organizations as socially constructed agents in the agent oriented paradigm. In: LNAI n. 3451: Procs. of ESAW'04, Berlin, Springer Verlag (2004) 1–13
9. Pacheco, O., Carmo, J.: A role based model for the normative specification of organized collective agency and agents interaction. Journal of Autonomous Agents and Multi-Agent Systems **6** (2003) 145–184
10. Giret, A., Botti, V.: Towards an abstract recursive agent. Integrated Computer-Aided Engineering **11** (2004) 165–177
11. Hauriou, M.: La théorie de l'institution et de la fondation: Essai de vitalisme social. Cahiers de la Nouvelle Journée **23** (1925)
12. Miller, S.: Social Action: a Teleological Account. Cambridge University Press, Cambridge (2002)
13. Tummolini, L., Castelfranchi, C.: The cognitive and behavioral mediation of institutions. Congnitive System Research **7** (2006) 307–323
14. Uschold, M., King, M., Moralee, S., Zorgios, Y.: The enterprise ontology. The Knowledge Engineering Review **13** (1998) 31–89
15. van den Broek, E.L., Jonker, C.M., Sharpanskykh, A., Treur, J., Yolum, p.: Formal modeling and analysis of organizations. In: Proceedings of the AAMAS Workshop on Organizations to Organization-Oriented Programming (OOOP). (2005)
16. Strang, D., Sine, W.: Companion to organizations. In Baum, J., ed.: Inter-Organizational Institutions. Blackwell (2000)
17. Galaskiewicz, J.: Interorganizational relations. Annual Review of Sociology **11** (1985) 281–304
18. Sannella, D., Tarlecki, A.: Essential concepts of algebraic specification and program development. Formal Aspects of Computing **9** (1997) 229–269
19. Sannella, D.: Algebraic specification and program development by stepwise refinement. In: Proc. 9th Intl. Workshop on Logic-based Program Synthesis and Transformation, LOPSTR'99. Volume 1817 of Lecture Notes in Computer Science., Springer (2000) 1–9
20. Jonker, C.M., Sharpanskykh, A., Treur, J., Yolum, p.: Design operators to support organizational design. In Gero, J., ed.: Proceedings of the Second International Conference on Design, Computing, and Cognition (DCC'06). Lecture Notes in Artificial Intelligence, Springer Verlag (2006)
21. Hübner, J.F., Sichman, J.S., Boissier, O.: A model for the structural, functional, and deontic specification of organizations in multiagent systems. In: Proceedings of the 16th Brazilian Symposium of Artificial Intelligence (SBIA'02). (2002)

22. Searle, J.R.: The Construction of Social Reality. The Free Press, New York (1995)
23. Searle, J.R.: Speech Acts: An Essay on the Philosophy of Language. Cambridge University Press, Cambridge (1969)
24. Masolo, C., Vieu, L., Bottazzi, E., Catenacci, C., Ferrario, R., Gangemi, A., Guarino, N.: Social roles and their descriptions. In: Ninth International Conference on the Principles of Knowledge Representation and Reasoning, Whistler Canada (2004)
25. Bottazzi, E., Ferrario, R.: Preliminaries to a DOLCE ontology of organizations. International Journal of Business Process Integration and Management (forthcoming) (2007)
26. Boella, G., van der Torre, L.: Regulative and constitutive norms in normative multiagent systems. In: Ninth International Conference on the Principles of Knowledge Representation and Reasoning. (2004)
27. Jones, A.J., Sergot, M.: A formal characterisation of institutionalised power. Journal of IGPL **3** (1996) 427–443
28. Governatori, G., Gelati, J., Rotolo, A., Sartor, G.: Actions, institutions, powers. preliminary notes. In Lindemann, G., Moldt, D., Paolucci, M., Yu, B., eds.: International Workshop on Regulated Agent-Based Social Systems: Theories and Applications(RASTA'02). Volume 318 of Mitteilung., Hamburg, Fachbereich Informatik, Universität Hamburg (2002) 131–147
29. Colombetti, M., Fornara, N., Verdicchio, M.: The role of institutions in multiagents systems. In: Workshop on Knowledge based and reasoning agents VIII Convegno AI*IA, Siena, Italy (2002)
30. Smith, B., Zaibert, L.: Hart, rawls, and searle: Social ontology and the problem of normativity. In Tsohatzidis, S.L., ed.: Intentional Acts and Institutional Facts: Essays on John Searle's Social Ontology. Springer (2006)
31. Hart, H.L.A.: The concept of law. Clarendon Press, Oxford (1961)

# Emergence In the Loop: Simulating the two way dynamics of norm innovation[1]

GIULIA ANDRIGHETTO
ROSARIA CONTE
PAOLO TURRINI
*Institute for Cognitive Sciences and Technologies, National Research Council (ISTC-CNR)*

**Abstract.** In this paper we will present the EMIL project, "EMergence In the Loop: Simulating the two-way dynamics of norm innovation", a three-year project funded by the European Commission (Sixth Framework Programme -Information Society and Technologies) in the framework of the initiative "Simulating Emergent Properties in Complex Systems". The EMIL project intends to contribute to the study of social complex systems by modelling norm innovation as a phenomenon implying interrelationships among multiple levels. It shall endeavour to point out that social dynamics in societies of intelligent agents is necessarily bi-directional, which adds complexity to the emergence processes. The micro-macro link will be modelled and observed in the emergence of properties at the macro-level and their immergence into the micro-level units. The main scientific aim of the EMIL project is to construct a simulator for exploring and experimenting norm-innovation.

**Keywords**: norm innovation, emergence, immergence, simulation, social complexity.

## 1. Introduction: an overview of the EMIL project

This paper introduces and illustrates the theoretical underpinning and the research agenda of the EMIL project, "EMergence In the Loop: Simulating the two-way dynamics of norm innovation", a three-year project funded by the European Commission (Sixth Framework Programme -Information Society and Technologies) in the framework of the initiative "Simulating Emergent Properties in Complex Systems", involving six Partners:

1. Institute of Cognitive Science and Technology, National Research Council CNR-ISTC Italy
2. University of Bayreuth, Dept. of Philosophy UBT Germany
3. University of Surrey, Centre for Research on Social Simulation UNIS United Kingdom
4. Universität Koblenz-Landau, KL Germany
5. Manchester Metropolitan University, Centre for Policy Modelling MMU United Kingdom
6. AITIA International Informatics Inc. AITIA Hungary

[1] This work was supported by the EMIL project (IST-033841), funded by the Future and Emerging Technologies programme of the European Commission, in the framework of the initiative "Simulating Emergent Properties in Complex Systems".



Dagstuhl Seminar Proceedings 07122
Normative Multi-agent Systems
http://drops.dagstuhl.de/opus/volltexte/2007/907

The current project will greatly benefit from the common activity of this consortium in the Multi Agent Systems (MAS) and Agent-Based Social Simulation (ABSS) fields, which provide a common methodology, vocabulary, way of modelling and observatory for the partners to collaborate towards the achievement of the task objectives.

EMIL is aimed at understanding and developing design strategies able to cope with particular types of complex entities, i.e. social systems. These are characterized by a two-way dynamics, consisting of emergent and immergent processes: emergence from interaction among individual agents, and immergence of entities (norms) at the aggregate level into the agents' minds. A summary of the main theoretical goals is:

- to understand and manage complexity in social systems with autonomous agents;
- to understand how new conventions and norms emerge, innovate, and spread in these systems;
- the study of norm innovation by means of agent-based simulation.

The main technological aim of the project is to construct a simulator for exploring and experimenting norm-innovation. The forecasted impact of the project is to contribute to the regulation of e-communities by handing out a simulator for the emergence of new norms in complex social systems.

EMIL appears as an important scientific opportunity, as it is a simulation project on complex social systems and the object of investigation being both emergence processes and immergence processes, which are peculiar to complex social systems. The process of immergence has not been deeply analyzed yet, and only from a theoretical point of view. EMIL shall provide a more careful, simulation-based investigation of this process. Through the analysis of normative innovation, the EMIL project aims to shed light on the insufficiency of a strictly conventionalist approach to norms and on the resulting need of a theory of (social, legal) norms, where an accurate distinction between norms and conventions is made. As the core of the matter is the way norms not only emerge, but also are innovated, it is clear that norms cannot be considered as conventions, i.e., mere behavioural regularities cannot be innovated.

## 2. Necessity for a scientific theory of norm innovation

In this section we shall endeavour to clarify some of the fundamental claims and background hypotheses of the project. We will also attempt to provide a wider theoretical perspective, mapping out a state of the art on norm innovation.

In complex social systems, norms and institutions have a short life. Although there are some primordial precepts, like reciprocity and word-keeping, finder-keeper and truth-telling, both legal and social norms often decay rapidly, soon to be replaced by new ones.

This is certainly the case with customary norms and conventions, such as greeting rituals. But it also occurs with legal or institutional norms, which undergo a continuous process of revision and innovation. Norms and institutions often prove inadequate and inconsistent, thereby leading agents to face and try to solve conflicts, assign relative priorities to clashing norms and avoid inconsistencies. This is done either by adjusting to them or elaborating contrary-to-duty obligations – that is how deontic logicians call 'the defective application of norms'. Heavily relying on their social intelligence and competence, individual social agents are expected to harmonise with and restore social order.

We define norm innovation as a particular kind of norm insurgence, i.e., an intended one. This does not necessarily mean that the source of innovation is personal, or that innovation concerns only legal norms. Indeed, we claim that norm innovation implies somebody's will to introduce and spread it. The process of norm innovation poses a number of fundamental scientific questions. These include the mechanisms by which new norms are conceived of; the conditions under which they spread; the extent to which they evolve as they spread through a society; the circumstances under which they become institutionalised and the processes through which they decay and are ultimately lost. There are also questions about the relationship between new norms and the context in which norm innovation takes place, for example, how new norms relate to existing ones.

EMIL is intended to find answers for these issues, obtaining significant advances in the scientific understanding of, and possibly in predicting, a fundamental aspect of social complexity, i.e. norm innovation, by means of computer simulation of complex social systems.

### *2.1 Background hypothesis: Emergence in the loop*

In this section, we will describe at some length a specific dynamic of the micro-macro link, crucial for explaining the process of norm innovation, i.e., emergence and immergence loop.

Complex systems are usually defined as composed of many different interacting elements, with non-linear interactions and network structure [32]. While this definition covers a wide range of systems, the EMIL project will focus on social systems only, composed of many interacting intelligent autonomous agents. Autonomous agents are able to adapt to, and evolve with, a changing environment, taking autonomous decisions. Moreover, autonomous agents with sufficient intelligence, such as humans, form new mental objects and processes consequent to the emergent behaviour of the system they are a part of, and act on the basis of these objects and processes. This is yet another loop that adds to the multiple chains of dependencies typical of complex systems, and is present in social systems uniquely because they result from the interaction of cognitive systems, endowed with intentions and other mental states. In these cases, we talk of social complexity: when change happens at some level, in order for the new pattern to spread over the whole system, some further modification is required in agents' behaviours and beforehand in their minds. This mental change allows agents to modify their behaviour accordingly; we call this phenomenon *immergence* [6]. The macro level and the (possibly many) micro-levels co-evolve while being connected by causal links in both directions, emergent and immergent, thanks to the mediation of the agents' minds. This leads to defining social systems as a special case of complex systems. Social complexity adds to general complexity the specific nature of social intelligent systems, the behaviour of which is regulated by internal, mental states. Hence, innovation in social systems is a bidirectional process:

- bottom-up: emergence of new entity or phenomenon (e.g., a given behavioural regularity, or agent property) at the aggregate level from interaction among agents;
- top-down: immergence of the entity or phenomenon in the minds of the agents, i.e. insurgence in their minds of a new mechanism, representation or process that leads the agents to modify their behaviours in conformity with the emerged effect. This is the key ingredient for the analysis of complex system including autonomous agents.

To be able to understand and manage socially complex systems, innovative approaches such as social, and more specifically agent-based, simulation appear to be essential for any serious scientific progress (for a discussion of when and why the agent base is vital for social simulation, see Gilbert, 2004). To avoid the pitfall of oversimplification, simulation approaches must take into account both the emergence of novel properties in the system, and the agents' immergent reaction to this.

*2.2 State of the art*

On the one hand, normative emergence has been the object of a wide amount of studies in several different academic disciplines. Apart from law sciences, there are also interesting contributions from Philosophy, Game Theory and Economics. On the other hand, norm innovation is a subject still not widely investigated and references are scanty if any.

In the scientific literature, two major views of norms co-exist: the social philosophical tradition [23] that equalises norms to conventions spontaneously emerging in a population, and the complementary notion of legal norms, offered by the deontic tradition and by the philosophy of law. Unfortunately, these two views have never been satisfactorily integrated [11, 12]. To questions like "where do institutional artefacts such as norms come from and how do they evolve?", the two traditions give totally different answers. For philosophers of law, norms are issued by central authorities. Within game theory, social norms are essentially seen as conventions, that is, behavioural conformities that do not presuppose any explicit agreement among agents, and emerge from their individual interests [23; 27; 28]. The function of these norms is essentially one of permitting or improving co-ordination among participants. Therefore, conventionalists identify norms with cooperation and fairness, finding their source in repeated interaction, or in reputation [22], etc. A convention gradually emerges from interactional practices, establishing who should do what. So, the emergence of cooperative (and fair) behaviour as a norm itself is driven by a particular order of preferences (in which strong emphasis is laid on the preferences defined as "social") [5].

Although this approach has significantly improved the possibility for game theory to describe and predict the behaviour of players, some preference changes are still obscure. For instance, the theory that fairness evolves among egoists as a rational solution to the problem of reciprocity [3] certainly accounts for an important function of social norms – namely, that of ensuring reciprocity in exchange – but it does not tell us how those norms evolve and spread, whether each agent separately "discovers" the advantage of fair moves or instead decides to conform to external pressures that converge on fair moves, and if so for what reasons, through which representations, etc.

With regard to the social side of norms, we auspicate a unitary view bridging the gap between conventions and norms. This, among other things, will allow to shed light on the question of norm-innovation. The conventionalist view can easily account for the proliferation of social rules, but it finds harder to explain how these can change, other than by mutation.

Indeed, hitherto, the only bridge between conventions and norms has been found in sanctions. As both conventions and norms are costly behaviours, they provide a rational incentive for

violation. To compensate or neutralise this incentive, they are often associated with sanctions for transgressors [4]. However, this solution raises other problems:

- How can we distinguish between norm-based sanctions from mere coercion? How can we model agents that are able to tell (and take into account) the difference?
- While rendering norm-abiding rational, the administering of sanctions increases the global cost of norm-based social order. It is appropriate, then, to explore the issue of cost distribution in norm-based social order. To preserve equity, some mechanisms should apply to ensure a redistribution of the costs of normative compliance. Plausibly, the decision to comply with norms in mixed populations, where norm observers may happen to interact with cheaters, could be strongly disadvantageous for individual norm observers.

The same intuition is at the basis of the economic interest and game theoretical approach to norms. In economics, at a first glance, the idea of complete contract was introduced in order to permit an efficient conclusion of transactions, always challenged by interest conflicts between parties [8]. Making institutions as part of this process has led to improving the efficiency of some interactions and to the completion of economic transactions. Nevertheless, on the one hand, this introduction proved of poor utility in those cases in which complete contracts were not possible [21], and, on the other, it hindered a better interpretation of most kinds of social interactions in real life – in particular, the most problematic ones, such as those needing coordination, or public good provision and, more generally, collective dilemmas. Hence, we need an operational and computational theory of norm-innovation, which gives norms a specific space, between mere conventions and pure coercion.

By importing and reinterpreting contributions, definitions, mechanisms, etc. from the social and legal fields, also computer scientists have (re)discovered the wheels of social order. And, as the problems mentioned above have been haunting social scientists for the last two or three hundred years, they now begin to trouble the minds of computer scientists. The issues related to the regulation of Multi-Agent Systems (MAS) are far more numerous and complex than could have been expected even a few years ago. When norms started to circulate in the field of MAS, they were seen as a concern of people working in the legal domain. But things have rapidly changed. The amount of MAS work on norms and institutions is on the increase. Whereas the conventionalist view of norms has inspired the work of Shoham and Tennenholtz [21], examples of the complementary view are countless [6, 10, 15]. Moreover, the reasons of interest, and the answers and solutions provided, have changed in a qualitative sense. Norms and institutions are now of

concern to scientists and designers of multi-agent systems and electronic societies, and more generally to scientists, designers, and managers of information societies.

Agent-based simulation of normative behaviour can be considered as a part of social simulation and, then, as one of the most promising tools in order to address some unsolved issues in the social sciences. Social simulation started to have a significant impact for social sciences in the 1990s [20], even though a few famous attempts can be dated earlier [28], and to our knowledge is the only instrument apt to deal with social complexity.

Finally, norms can be considered as the ideal example of a complex social artefact, because of their role in connecting emergence and immergence. Intelligent agents, which act (at the local level) on the basis of intentions and other mental states, can produce change at the system level. Recognition of system level changes, and reaction to them (immergence), is an option available to intelligent agents only. To make immergence possible, agents must create and utilise specific representations in the form of cognitive artefacts: the main example being norms. Thus, norms can act as a pivot connecting emergence and immergence, located at the intersection between the micro-to -macro and the macro-to-micro processes.

## 3. EMIL theoretical, technological and application challenges

In this section, we shall endeavour to give an account of the main EMIL theoretical, technological and application challenges.

### *3.1 Theoretical*

As we have illustrated, the EMIL project is aimed at providing a theory of norm innovation (EMIL-T) by means of agent-based simulation, understanding not only how new conventions and norms emerge, but also how they immerge in the minds of autonomous agents.

To understand norm-innovation, the project needs to model the 2-way dynamics of sociality (EMIL-M), consisting of emergent processes [14, 17] and immergent effects [6, 12]. In particular, the two-way dynamics of social complexity allows a number of questions to be highlighted, that can be listed as follows:

- What are the factors (both inter and intra agent) that give birth to the emergence and diffusion of new conventions in a complex society (by convention essentially meaning a behavioural regularity observed in a given population at a given time)?

- What are the conditions that favour the stabilization of new conventions?
- What is the difference, if any, between a convention and a norm? Intuitively, a norm is a legitimate *prescription*. What are the objective and immergent correspondents of this notion [10] and how to operationalise it?
- How do new norms relate to existing ones? More generally, what is the relationship between new norms and the context in which norm innovation takes place?
- Finally, when and why does a norm become useless and/or non-prescriptive, when does it come to decay and, in the end, disappear?

Both theoretical and applicative tasks of the EMIL project are directed to provide an answer to these questions.

### *3.2 Technological*

A theory of social order must be bi-directional [9, 33] and show how innovative phenomena (e.g., institutions) affect the social systems they emerge from.

While research on the construction of interacting artificial cognitive agents is not a new area [30], current systems still lack the scalability necessary for large scale modelling. For this reason the main technological aim of EMIL is to construct a system that is capable of performing norm innovation at the appropriate scale (EMIL-S). In order to build this system, we will need to

- develop a platform for simulation of cognitive agents using appropriate Open Source and readily available technologies;
- integrate simulations on different agent levels (e.g. agent, institution, and society), modelling the interplay among different levels in norm innovation;
- integrate data acquisition and modelling, by taking into account data from simulation while modelling and building the simulator.

### *3.3 Application*

The EMIL project is expected to contribute to the regulation of e-communities by handing out a simulator for the emergence and innovation of new norms in complex social systems, whereby situated experiments can be run. While the simulator will be designed as a general-purpose tool, some specific case studies will be selected so as to provide the necessary grounding parameters. The selected field of application will be the rise of collaborative community norms in the Open Source

community, where new conventions have been established and new norms are being invoked. The stabilization of both is a major concern in this community, which offers (a) an interesting observatory of the mechanisms and processes of norm innovation, (b) an appropriate environment from which to draw essential data to be fed into the EMIL Model (EMIL-M) and Simulator (EMIL-S), and (c) an application field for testing its utility.

## 4. Relevance

EMIL is intended to provide both theoretical and implementative advances in those fields of critical information infrastructures that are of crucial interest for economy, as they can improve the competitiveness of businesses and firms, and for society at large, as they can provide a sound and multi-purpose tool for policy making, norm enforcement and trust enhancement.

In particular, it is concerned with norm innovation, an issue of growing concern in natural, electronic and artificial societies. In order to figure out the social impact of the EMIL project, suffice to think of the variety of concrete phenomena and examples to which norm innovation applies. A social, institutional, technological and natural environment in constant and rapid transformation requires an equally assiduous updating of policies and customs, habits and conventions. The necessity for tools and instruments allowing the temporal, structural and behavioural conditions for norm innovation to be detected, tested and predicted, thus becomes transparent.

Furthermore, attention paid to norm innovation is on the increase in several fields of Information and Communications Technology (ICT), and is strictly intertwined with software development and diffusion. This is certainly the case in the field of agent systems - e.g., in teamwork and robotic applications to domestic work and assistance - where the necessity to bring together human and software agents, in hybrid systems of interaction and advanced technology soon became clear. In other ICT settings, norm-innovation is perhaps an even more urgent necessity. In e-markets and businesses, as well as in e-communities like those formed around the Open Source, the problem of trustworthiness is widely perceived as the problem “number one” for a decisive consolidation of the new tools. In turn, trustworthiness requires new conventions and social norms being brought about for governing and at the same time giving impulse to complex agentized environments [13, 24]. To drive innovation and growth in agent applications requires the invigoration of the problem of social order in the context of the new economy and society. On one hand, a scientific, explicit and operational theory about what type of norms will emerge under which conditions and to what extent is much wanted. On the other hand, for the assumption of

social complexity as formulated above, this requires a theory about how, when and to what extent any given norm, once emerged, will have impact on the agents' behaviour. For the assumption of autonomous agency, this implies that we develop a theory of how a norm is implemented in the mind.

After centuries, the problem of social order is still waiting for a theory that accounts for similarities and differences among different types of norms (conventions, social and legal norms), as well as for their prescriptive force, and that establishes a clear-cut confine between norms and mere coercion (which is also based on sanctions). Whereas the emergence of conventions has received considerable attention, the way-down in the circuit of emergence has been accounted for, if only, in purely behavioural terms. Hence, the process of immergence has being generally overlooked, and with it, the emergence of norms as something in connection with, but distinct from, mere conventions and coercion.

A two-way theory of norm innovation is a necessity for the technological advance of society and a challenge for science. Moreover, or more generally, no development of the science of complexity in the direction of social matters is possible without accepting and taking into account the assumption of social complexity.

Our objectives, both theoretical and technological, will be pursued throughout the project, by building a solid theoretical framework to be used for modelling norm innovation as a two-way social dynamics, by applying this model to concrete empirical scenarios, and by developing simulations, integrating data and revising the theory and the computational tool on the grounds of the findings obtained.

Our project could not be carried our without developing a model of emergence of aggregate behaviour, considered both at the macro-level (the emergent property) and at the micro-level (mental objects and processes agents develop as a consequence of the emergent property, guiding their actions in a complex environment). This model is aimed at simulating and experimenting upon the conditions favouring the immergence of a given emergent property (a norm), thereby allowing for the process of norm innovation to be checked and possibly predicted.

## 5. EMIL potential impact

EMIL's expected impact is proportional to the increasing necessity, within the IT field, for regulated e-societies and communities, and consequently for norm-innovation. Hence, the impact of the project is proportional to the achievement of its objectives; in particular, the building of a platform for simulating multi-agent interaction is expected to allow the conditions under which

given norms emerge, innovate, and spread to be explored and experimented upon; hence, conditions favouring the insurgence and stability of such norms might be predicted and fostered. Indeed, one of the main problems faced by designers of advanced technologies of interaction in different domains is the need for distributed regulation, i.e. conventions and social norms evolving and spreading with none or reduced need for centralised and deliberate intervention. The strategic impact of EMIL essentially lies in its contribution to predict and promote the regulation of e-communities by means of new norms and conventions. A simulator of norm innovation in a society of artificial agents endowed with variable autonomy and intelligence will be developed.

The setting to which the proposed project is inspired is the Open Source, as a scenario in which innovation is demanded and appears to depend on both theoretical and technological advances. Although the Open Source community has made substantive advances in the last few years, it is now about to cope with a major challenge, i.e. reduce proliferation of licenses on the one hand and tackle the issue of deployment. While technical problems are surmountable, the emergence and acceptance of regulation, conventions and standards seem more questionable and are strongly dependent on a scientific understanding of how the emergence, innovation, and spreading of norms and standards are conceived of. To be noted, the competitiveness of the Open Source is highly related to its community coping with this challenge. To quote Michael Tiemann (in his speech at the O'Reilly Open Source Convention, 2001), “this issue is important because it is about the future of software (the increasing substance of technology), it is about the increasingly important aspect of technology as it relates to our economy, and it is also about the code-as-law that will ultimately govern us”.

Indeed, Open Source not only means access to un-obfuscate source code, but also conformity with a number of criteria, such as free redistribution (license shall not require a royalty or other fee for sale); integrity; no discrimination against persons or groups, nor against fields of endeavour (be it a business, or genetic research); distribution of license (to forbid closing up software by indirect means such as requiring a non-disclosure agreement). Furthermore, license must not be specific to a product, nor should it restrict other software. Finally, licenses must be technology-neutral. The future of advanced technology and, for that matter, the future of scientific discovery is somehow conditioned to the destiny of Open Source. Writing good software with sensible licensing terms encourages a better, more transparent, more trustworthy architecture for computing, it empowers individuals, promotes free and equal competition, and enables freedom at higher levels so that others can build applications with confidence. Indeed, the Open Source community has learned that rapid evolutionary process produces better software than the traditional closed model, in which only a very few programmers can see the source and everybody else must blindly use an opaque block of

bits: "…when programmers can read, redistribute, and modify the source code for a piece of software, the software evolves. People improve it, people adapt it, people fix bugs. And this can happen at a speed that, if one is used to the slow pace of conventional software development, seems astonishing." (Official site of the Open Source Initiative, http://opensource.org/index.php)

On the other hand, the Open Source development has led to a proliferation of new licenses, which in turn represents a significant barrier to open-source deployment. Let us see why. Open-source software has developed to a degree that would have been unimaginable a decade ago. Governments around the world have responded to the open-source message with initiatives and funding. Our entire economy and the society around it has benefited as the new and larger open-source community gave software consumers and technology users everywhere a whole new range of choices.

However, there are three problems around software, i.e. development of high-quality, innovative and reliable products; its distribution; its deployment, i.e. the management of the technical and legal complexity of software in use.

The first problem is no more a serious concern. That open-source developers out-compete proprietary software in innovation is no more news, although the interesting question is why, to what extent and under which conditions this is the case.

As to the second problem, it must be said that open-source licensing and the explosive growth of the Internet has combined into a mutually-reinforcing attack on software distribution costs. Hence, even distribution is no more an issue.

However, solutions to the development and distribution dilemma create a problem of deployment. Why? Because the central activity of the open-source community is to create, re-use, and re-combine source code. Combination can leave software developers, users, and distributors uncertain as to their rights and responsibilities. Uncertainty makes people afraid and prevents them from making any move; thus raising the costs of using Open Source; and injures everyone. The problems can indeed be solved, as technical difficulties are surmountable. As usual, the real problems stem from social cognitive aspects. A change is needed in the criteria used to approve licenses. The license should be original, solving a problem not sufficiently addressed before, and must be clear and reusable. In particular, as to reusability, licenses that tend to create ghetto-ed communities attached to a single firm or vendor are discouraged. Corporate players ought to be let to observe the benefits of giving up proprietary control gradually, rather than asking them to give up all control immediately thereby pushing them into a defensive position. Since licenses reduce the growth of development communities, corporate players that originally promoted them now complain about the proliferation of licenses and the legal complexity created by license collisions.

Therefore, the best way to serve corporations and the entire open-source community is to insist on reusability without qualification and a general removal of legal barriers. The new criteria might annoy and even anger some people, and this is a necessary evil, if one wants the proliferation problem to be addressed seriously. But how to encourage search for approval and acceptance of new criteria?

To develop an anti-proliferation policy means to develop a scientific understanding of the standards and of the criteria there under, as well as of the processes and conditions leading to people's acceptance of standards and their decision to meet new criteria.

## 6. EMIL Workplan

EMIL is articulated in three main building tasks, EMIL M, EMIL S, EMIL T, and in two testing tasks aimed at collecting real data and simulation findings: one task dealing with the Open Sources scenarios, the other with the simulation executions. In this section we summarise the EMIL workplan, pointing out the main scientific and technological challenges, the focal articulations and the links between the tasks.

### *6.1 EMIL-M: Theoretical background on norm innovation*

EMIL-M works out a general *model* of norm-innovation a complex social dynamics among intelligent social agents, to be tested by means of agent-based simulation in a specified context.

The case of the emergence of norms in Open Source will be analyzed in another task of the project (one of the two testing tasks) in order to provide empirical examples of the emergence of such norms contributing to the construction of a grounded EMIL-M. Open Source here refers not just to Open Source software, but also to other domains where participants freely provide intellectual goods to others, outside a market relationship. EMIL-M will raise questions that will be answered in the EMIL-S and EMIL-T: what is the difference between a norm and a convention? How is it possible to pass from a norm to a convention, or the other way round? When and why will new norms arise? Can we envisage and hypothesize on the conditions under which norm innovation is likely to stabilise and spread? Is it possible to derive a typology of norm innovation, so as to understand the cognitive structures involved?

In order to develop EMIL-M, both conceptual clarification (working out vague concepts, imprecise definitions, unsolved problems) and model development are needed, with the main goal of building formal theories that can be used for simulation. Since norm innovation results from a

complex collection of nested theoretical definitions, it will be useful to provide a shared ontology, or in other words, to forge a common vocabulary of interrelated notions with which to work. By ontology we mean a conventional and operational tool, a set of theoretical notions that are defined one in relation to the other. Its goal is to make conceptual links explicit. It is the starting point of the project and provides shared conceptual and theoretical instruments (ontology about norms as well as emergence and immergence theory) for executing the project workplan.

In the picture below, you can see a provisional schema of the shared ontology that we intend to model.

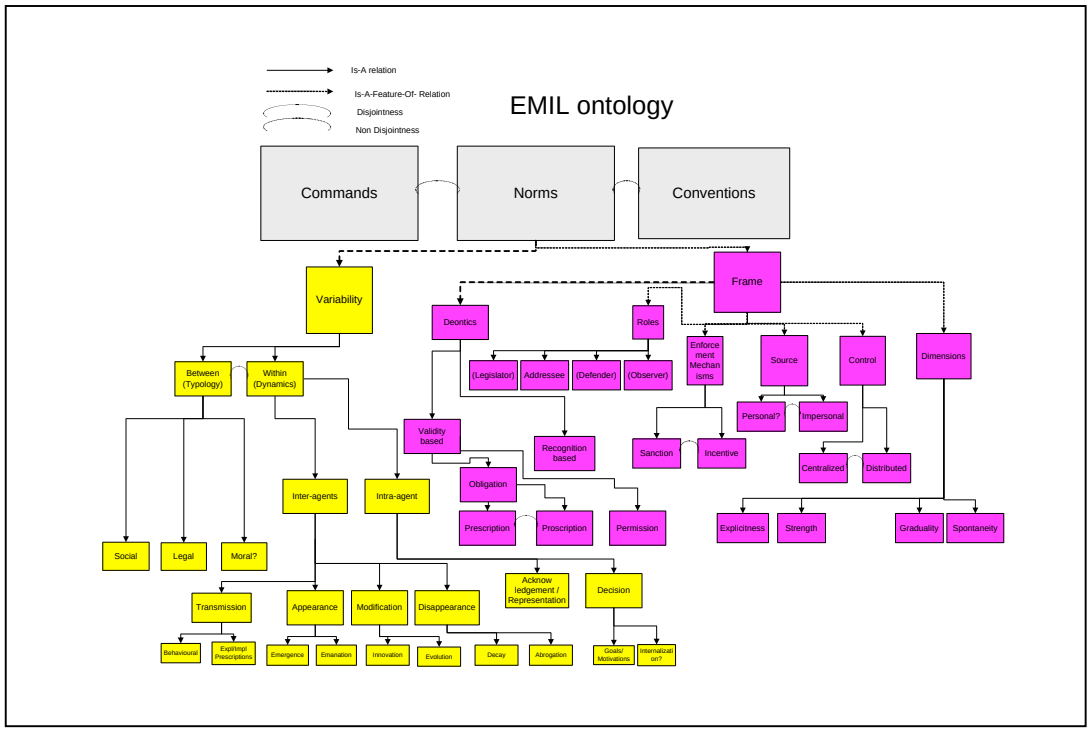



Fig.1: A provisional schema of the EMIL's ontology

As you can see in Figure 1, the first three entries in EMIL's ontology are commands, norms and conventions, notions that constitute the theoretical kernel of the EMIL project. As these concepts can have different definitions in different theoretical fields, we will try at first to find a shared meaning for them, endeavouring to shed light on their similarities and differences. We will then focus attention on the notion of norm, decomposing it in its most important features (that are themselves decomposable), in such a way that it becomes more clear and useful for the aim of the project.

Apart from the shared ontology, EMIL-M will consist of an analysis of the so called inter agents processes and intra agent properties. On the one hand, inter agents processes contribute to

characterize the social side of the norm, on the other hand, intra agent properties define the internal side. For the inter-agents processes, special attention will be given to the mechanisms of emergence and diffusion of entities or properties at the aggregate level, from interaction among agents. For the intra-agent ones, the attention will be focused on the mechanism of immergence.

Autonomous agents living in a complex environment are endowed with skills allowing them to modify their world. Their actions not only affect particular states of the world, but also other agents' mental states and actions. Emergent properties are those properties that (a) emerge from interaction among agents and between the agents and the external world; (b) have particular and unpredictable characteristics not deducible from properties of entities at the original level, and (c) produce a change in the overall system.

Emergence is quintessential for the definition of complex systems. However, as we deal with autonomous social agents, emergence is in the loop between bottom-up and top-down processes: emergence of properties at aggregate level cannot be accomplished effectively unless properties feedback on the lower level through a complementary process of immergence into behaviours of units (agents) at the lower level, and beforehand into their minds.

Speaking now of the diffusion of these entities and properties at the aggregate level, we must observe that sociality is not independent of cognition. Norms drive behaviour once they have accomplished the immergent process. Therefore, with cognitive agents, behavioural diffusion of norms can only be predicted on the grounds of the effect of norms on agents' mental processes. Diffusion has to be studied to understand what are the conditions according to which both norms and convention spread and stabilize, and what are the differences between these two categories.

As noted before, there are two main processes adding to the complexity of social systems. On the one hand, entities (like norms) emerge from interaction. On the other hand, agents are affected by these entities (immergence: the effect of emergent properties on the agents' minds) in two consecutive steps: at the mental level and at the behavioural level.

Consider, however, that although the former step is necessary for the latter to be executed, it is not sufficient, since mental immergence can result in norm violation. Immergence makes interaction much more complex and unpredictable, and gives rise to new emergent properties at different ontological levels. Norms emerge from social interaction but agents reason on them and act according to their beliefs; thus, norm diffusion and stabilizing are strongly affected by cognitive processes. For this reason, it is an urgent task to clarify some components of the mental processing of norms.

First of all, a norm becomes a belief in the mind of its subjects, namely the belief that a given

behaviour, in a given context, for a given set of agents, is forbidden, obligatory, permitted, etc. More precisely, the belief should be that "there is a norm prohibiting, prescribing, permitting...". Indeed, norms are aimed at and issued for becoming such beliefs. In other words, norms must be acknowledged as such in order to properly work; this is their function [10, 12]. In order to realize the existence of a norm and its impinging on somebody, the belief that to do something is prescribed or forbidden is not enough. The norm is something more than a mere will, an order of a private agent that obliges us to do or not to do something. The binding force is insufficient, since it characterizes also non-normative commands. The point is the source of the norm, the issuer of commands. For a prescription to be perceived as a norm, who should the source be, how should it be characterized? EMIL-M shall attempt to give an answer to this crucial question, endeavouring to characterize the normative source, i.e., the nature of normative authority.

Believing that a norm exists and concerns us requires at least a second group of beliefs: the beliefs of concern. The norm says what ought to be done by whom: (i) the obligation/permission/prohibition and (ii) the set of agents on which the imperative is impinging. For example, if I am addressed by a given norm (say, "be member of a professional order"), and the norm has to take effect on me, I must recognize this. The prescription is about a set or class of agents, and since I am an instance of that class, the norm applies to me.

However, a normative belief is not yet sufficient to yield a norm-governed behaviour. A complex mental representation is needed, which includes several types of beliefs (normative belief, normative belief of concern), goals (normative goals), meta-goals, and rules of normative reasoning (norm adoption) [10, 12]. In this sense, cognitive ingredients have to be carefully taken into account, properly analyzed and modelled, especially concerning the properties at an aggregate level that come into effect in the agents' minds, this is the main focus of the analysis of the intra agent dynamics. Along with emergence, immergence is the fundamental mechanism of the micro-macro link, whose complex dynamic allows for phenomena like norm-innovation, typical of those societies of agents that are endowed with cognitive ingredients. In order to be understood, norm innovation requires a theory of bidirectional process of both emergence and immergence.

One aspect of the theory of norm immergence still unexplored is the mental object or mechanism that is regulated and affected by the norms, i.e., the mental scope of norms. In other words, norms not only regulate behaviour, as supposed by a BDI architecture, but also act on different aspects of the mind depending upon the agent architecture, namely its beliefs, emotions and decision making.

For what concerns the norms' impact on beliefs, let us consider common obligations, as in "you OUGHT to believe what your mother says", or the more extreme obligation "You MUST believe in

God". Notice that although there is a behavioural side of belief acceptance, consisting in agents behaving as if they believed in God or another authority (such as parents), the norm expects not only to regulate the external behaviour but also the underlying mental states. Why should we adopt an obligation to believe something? Part of the prescription recommends that agents do trust certain figures, acknowledging their authority as an emanation of the authority of the norm itself, which expects to be honoured by the subjects. But it is unclear whether this is all we can say about this phenomenon, or there are also other ways in which one can be prescribed to believe.

With regard to the obligation to feel something, it may be interesting to investigate under what circumstances, you MUST feel ashamed, or guilty. The locution: "BE ashamed" appears as an impossible command, of the type "BE spontaneous". In real matters, it is much more efficient. Rarely, people that are told to be or even feel ashamed remain untouched, unmoved by it. Why is it the case, and how is it possible? Again, one might say that what is in the scope of the norm is not the feeling, but its behavioural expression, its open display. But this is not the case: agents cannot help experience a feeling of shame when ordered to do so. Another example is "You OUGHT to do your work out of vocation".

Finally, norms act also on decision-making, and any of these mental objects is decided upon to some extent, included emotions. Let us consider goals. An agent decides to comply with a norm that has been integrated in the agent's knowledge base, giving rise to some sort of normative goal. An agent endowed with this particular kind of goal is allowed to confront it with any other goals of his and, to some extent, to choose which one will be executed. The normative goal can eventually be abandoned, otherwise it becomes an external motivation that drives the agent to act. Sometimes, the decision to comply with norms can have an endogenous drive. In such cases, the norm is followed according to an internal reason, often associated to given emotions, such as shame, remorse, or feeling of guilt. In these circumstances, we say that the norm has been internalized.

To sum up, obligations are represented as specific reasons to believe, feel, want, act, etc. This is important for an integrated theory of the mind. What is not clear yet is whether agents in fact modify their mental states as a consequence, or simply know they ought to. In the latter case, what changes is the goal and the consequent action, with the norm acting as any other incentive. In the former case, the prescribed mental states do in fact get formed or modified as an effect of the normative belief. But how this can happen is still unclear and a more accurate model of immergence is one of the main theoretical challenges of the EMIL project. Together with a theory of norm emergence, it will represent the kernel of EMIL-M.

### *6.2 Open Source scenarios*

A great deal has already been written about the Open Source software (OSS) movement, and there are several projects devoted to various aspects of OSS within the Framework Programme.

This task will not duplicate this work; rather, it will draw on the data and results generated by these projects to develop a history of OSS that focuses especially on the emergence and innovation of norms within the OSS community. Ours is a perspective on OSS that has only been touched on by the existing projects and so, while they have created much data of great use to this task, there may also be a need to go beyond this secondary data to collect some primary data directly. For example, it may be desirable to interview some participants (probably using online interviewing by email or instant messaging, rather than face-to-face) or to re-analyse some of the records that are left for researchers to study by the operation of OSS (e.g. online forums, bug trackers etc.). What really matters by means of taking a new analytical perspective, in accordance with the theoretical model developed in EMIL-M on OSS movement, is to understand which fundamental factors have given birth to the emergence, innovation, and the diffusion of conventions and norms in our complex society. On the grounds of this, EMIL-M will also be translated into a more specific simulation model anchored to the data provided by the above scenario.

Although OSS is the most famous and the most studied example of Open Source, there exist many others and the number is growing. Academic science, although having some distinct characteristics, can be regarded as the 'father' of Open Source and it will be interesting to compare the development of OSS norms with the history of the emergence of the 'norms of science' which have also been studied in depth by historians, philosophers and sociologists of science. This points to the understanding of the emergence of new norms among existing ones. Therefore, it will allow for the analysis of the causes that have brought about the decay of such norms. There are also examples such as Wikipedia which have appeared only within the last couple of years - which should make tracking the history somewhat easier, since the investigation will be almost contemporaneous to the emergence of norms.

We will first collect potential candidates for study, specify the scope of the candidates and their answering the crucial question about norms dynamics, locate sources of existing data about each of them, make contact with prior or current projects that are studying these or similar examples, and end up with the choice of two (or three, depending on the richness of the data) of the examples for further investigation.

For each of the examples selected, a historical survey will be written, focussing on the

emergence of norms. The approach adopted in these histories will be strongly influenced by the theoretical approach adopted in EMIL M and will consider not only the ways in which norms (for example, of how one claims authorship of intellectual property) emerge from interactions between participants, but also the ways in which these norms are stabilised, diffused, innovated, and used by the participants to understand the complex domain and act on it. The histories created will be conventional prose accounts. They will be later transformed into a more computational form, so that they can be used to contribute to the modelling of EMIL-S phase as empirical test data for the simulation models.

### *6.3 EMIL-S: Simulator Building*

A simulation platform for experiments, EMIL-S, will be built up with the purpose of testing EMIL-M. EMIL-S will be applied to execute simulations and compare results with available data, especially in the field of Open Source. The simulation platform will be a software architecture, written in an object-oriented language, consisting of ready-made and extensible simulation programs, including software agents that can interact with each other, aimed at giving an account of a precise range of phenomena.

The simulator will provide templates for different kinds of agents (representing both individuals and aggregates) as well as different kinds of environments within which the agents will have to interact.

Agents will be able to keep memories of encounters with other agents and will also generate (incomplete) models of other agents and of their environments. The simulator will provide these technical characteristics of agents and their relations, as well as a protocol for the communication among agents; details about attributes of agents and relations and communication will be worked out in collaboration with the task in which the simulation execution will take place.

EMIL-S is one of the main scientific/technical challenges of the EMIL project. It will provide a research instrument for norm innovation, and will be released in the public domain for customisation and for application to additional experiments and domains; non-research users will be reached by the dissemination assists, and will range from policy makers to international organizations. While aiming to simulate the Open Source scenario, the simulator will be designed as a general-purpose instrument for the analysis of norm innovation. The design and realisation of EMIL-S will be characterized by the following four stages.

At a first step, it will be necessary to enact a thorough analysis of the requirements of the

simulator needed for the purposes defined in EMIL-M and in the task concerning the Open Source scenarios. These purposes have to be explicitly linked to answering the questions subtending the whole project (in short, what are the mechanisms by which new norms are conceived; the conditions of their spreading; the circumstances under which they become institutionalised; and the processes through which they decay?). An evaluation of available simulators (REPAST, MASON, Swarm) and agent models (BDI and similar, i.e., JADEX) will open this phase. In any case, we will need a customised user interface and several improvements over the current state of the art, especially for that which regards a) multi-level: agent that can be composed of other agents, and b) internal norm representation and reasoning.

The second step is the design of the simulator. It will consist of

- a formal description of the relation and environment classes as these will form the three main classes to be provided by the simulator, and of
- a graphical user interface which would allow the simulator to output simulation results in textual and graphical ways, the former to allow for further analysis with standard analysis tools (statistical analysis systems etc.), the latter to allow for graphical output of topographies, networks and time-series during the simulation run.

Moreover, we will design an agent architecture specialised in norm manipulation. Agents will be built in increasing levels of elaboration, starting from cellular automata to BDI agents with explicit beliefs and goals. In particular, the more elaborated agents will be able to:

- recognize the onset of conventions, and possibly norms;
- manipulate the factors that give birth and stabilize new conventions and/or norms;
- recognize institutions and interact with them;
- reason upon the interaction between different, possibly contradictory, norms;
- reason upon the possible decay and disappearance of norms.

The result of this part will be a formal description of the agents, in increasing levels of elaboration. Moreover, in order to make the simulation platform usable by the largest possible audience, a graphical user interface will be designed in order to allow users to derive their own agent classes from the templates pre-defined by the simulator.

In the third step, we will implement the design in a more or less platform-independent way and document the implementation in a way that allows users the extension of the simulator and its use

as a platform from which user-defined agents, relation and environments can be thus derived from the mentioned built-in templates. It will be necessary to have the simulator run in a distributed way, i.e. on any number of networked machines.

As of today, user-defined agent, relation, and environment classes will be defined in JAVA code (in earlier times it might have been necessary to design textual simulation description languages, but most modern development kits nowadays combine the graphical manipulation of symbols on the user interface with a description of details in one of the ordinary general-purpose languages such as JAVA, and this will also be the policy of the new simulator).

The fourth, and last step, will overview the development process in order to ensure the necessary quality level. In particular, the necessary testing protocols will be issued and tests will be performed. Procedures for software maintenance (bug report and reaction protocol, reaction to software component upgrades, inclusion of third-party future extensions) will be developed.

### *6.4 Simulations executions*

In this task a concrete implementation of an abstract model of norm-innovation in the Open-Source scenario will have been created.

The current task is responsible for creating and executing carefully designed computational experiments in levels of growing complexity. Simulation execution will be carried out with focus on parameter space exploration, efficient execution and accurate result analysis, setting the stage for the final theoretical advances (EMIL-T) and contributing to the refining of the simulation platform (EMIL-S).

Models of complex social systems typically depend on a number of assumptions, quantified in the form of specific values to certain model parameters. Ideally, any such model should be tested with any meaningful combination of these parameters, in order to determine the validity of the model.

In addition to the dimensionality and the size of the parameter space, the *sensitivity analysis* of complex system models has to face the additional challenge of establishing the results' statistical validity, independent of the probabilistic model elements. Because computer programs, and thus computational models, are inherently deterministic, random factors are modelled by using so-called pseudo random number generators (RNGs). RNGs generate a deterministic sequence of numbers, with the desired statistical properties, depending on an initial value termed *seed*. Different seeds result in different random number sequences. Thus, the task of establishing the results' statistical validity involves running the simulation with various RNG seeds and analysing the collected

results.

This task will be intended to underline the most relevant quantitative findings on norm innovation, pointing to potential breakthroughs in this field. Data to be collected need to comprise different levels of hierarchical structures (agents, institutions, societies) to point out interdependences and micro-macro dynamics of norm innovation.

It has to be said that running simulations at this scale and complexity is from a computational point of view a particularly intensive task. This creates a special emphasis on the design and sequencing of the experiments, so that they allow for branching depending on earlier results and also for revisiting previously explored areas of the parameter space with greater 'resolution'. Even in case of carefully designed experiment plans, the task may easily exceed the abilities of today's PCs or workstations. Therefore, the execution must be distributed among several computers on the network.

There are two basic approaches to do this. In the first, each simulation instance is run on a single computer, but the many instances required for parameter space exploration are distributed over the network. In the second, even components of the same simulation run (e.g., the agents) may be divided up among the participating computers.

In the former case, neither the size nor the computational requirements of any of the simulation runs to be executed can exceed the capabilities of the participating (single) computers. However, this approach is relatively easy to implement, and can speed-up the execution of the set of experiments (typically involving thousands or tens of thousands of single simulation runs). On the other hand, the latter case allows for the execution of simulation runs that exceed the capacity of any participating computer. However, this comes at a price of considerable implementation efforts on the side of the engineers of the execution platform, and typically also on the side of simulation developers. It also raises the issue of close synchronization among the participating computers, since discrete time models typically assume a single 'central clock', which is often inconvenient in a networked environment with computers with varying computational abilities, it can often be the case that the entire 'computational pool' operates at the rate of the weakest participating unit. Moreover, in case of models with strong interaction/communication among the agents, the second type of distribution is highly impractical due to the increased communications costs. On the other hand, in principle, the open-source scenario deals with agents in a highly distributed physical setting, with particular ways of typically asynchronous communications. Therefore, despite the general wisdom that ABM's typically benefit from the 'first type' of distribution only, as a first step of this task, knowing the exact model developed in EMIL-S, we will consider the option of utilizing the 'second type' of distribution, too.

The first step of this task will consist in assembling a pool of computers for simulation execution and to the automation of the execution of simulations on it. The automation part also includes collecting the generated results from the various computers, and assembling them into a single, coherent database.

Since the anticipated application will be likely to generate large quantities of data, a special consideration to data formats is required in order to ensure the efficient analysis of the results. The simulation execution system will be created using available technology and open-source software. These technologies include functions provided by *grid systems* [16], where a pool of networked computers, whose availability varies in time, is assembled, providing a single access point with a common, unified 'operating system shell' for users. While our execution system will definitely use a pool of networked computers, and while from the point of view of the simulation application, these computers will be accessed via a single access point, we will not need the whole functionality of a grid. This is mainly due to the closed nature of our system: we will not have many individual users.

Our system will be dedicated to the execution of a series of runs of the same simulation (at any given time) with different parameter combinations and to the distribution of these runs across the pool of available computers.

The second step is concerned with the design of the set of computational experiment, focusing on the search of a representative but manageable parameter space. Models of complex social phenomena should be tested with any meaningful combination of parameters, in order to determine the validity of the model. In cases where the model's output depends linearly on the initial parameter values, this task is easy to accomplish. However, this is typically not the case with complex social phenomena.

Experiments will extensively explore the parameter space of the model, concerning both the scale of the system and the internal complexity of the model components (i.e., the agents, institutions, and society). This will involve: (a) establishing the values of parameters and initial conditions (b) defining appropriate output indicators (c) running the simulation and comparing the output with data obtained from the Open Source scenarios task. The dependence of the outputs on the specific values of the parameters will be examined, by using an automated 'sweep' of the parameter space (either by exhaustive iterations - for a few parameters - or by random sampling, or both).

In this phase, the actual execution of EMIL-S based simulations will be carried out. This involves looping through this task and the next one:

- Execution of the simulations.

- (Statistical) Analysis of the results.

These steps will be iterated, incrementally collecting knowledge about the model's behaviour, gradually refining the exploration area based on earlier results. This kind of iterative approach is particularly important due to the expected non-linear nature of the behaviour. For example, the first set of runs might point out important disturbances at a certain range of parameters, which will be further explored in finer details in the subsequent simulations.

Finally, we will collect and analyse simulation results. This task will give feed back to the previous one, requesting for more simulation where answers will be statistically uncertain or accelerating parameter sweep where results appear to stabilise.

We will compile a report on the collected findings and provide practical and theoretical interpretations. This also involves evaluating the findings of the computer experiments, among others, by fitting the generated simulation results to the data collected in the Open Sources task.
In the case of any model, especially in the case of computer simulations, a question of paramount importance is the validity of computational investigation. Validity is always assessed in the context of a comparison. The output of the model may be contrasted to the present or future state of the modelled system, but also, to the output of another, pre-existing model, or to theoretical conclusions developed by other scientific means.

This means that validity can only be properly assessed after fixing the purpose of the computational model, which can be of several general types. Often, the model's future is compared to the future state of the target system. We call this *prediction*. In other cases, only similarity of the two systems' behaviour is required. We call this *simulation* (which term here, despite the same word, does not simply denote the execution of the computational model). A well-known example of this approach is Craig Reynolds' popular BOIDS (or flocking) model [26]. Finally, we talk about *thought experiments* when the model's conclusions are not directly compared to the real system, but rather to a set of accepted theories and their conclusions. For example, in its evolution of the cooperation model, Robert Axelrod generates an existence proof by agent-based modelling, showing that spontaneous cooperation can emerge given the generally accepted assumptions about rational, selfish behaviour [1]. In the case of the latter kind of models, scientific work is the collection of a set of sufficient conditions that lead to the emergence of the studied phenomenon.

In the case of the EMIL the goal of model development is two-fold. First, we aim at explaining the emergence and innovation of norms in complex social systems. Therefore, our simulation results must be compared to pre-existing theories and assumptions that will have been collected in the Open Sources task. The second aim is qualitative prediction of the emergence and innovation of

norms in the particular domain of our simulation, in the Open Source software development community [31].

### *6.5 EMIL-T: Integration and theoretical update*

After having constructed a model of norm innovation at the social and cognitive levels, described an empirical example of norm innovation (the development of norms in the open-source movement), and applied a computational version of the model to the empirical example by building and executing simulation experiments, EMIL-T shall evaluate the success of the model by comparing the results of the simulations with the empirical data documented in the open-source scenario. The revision and improvement of the theory of norm innovation (EMIL-T) represents one of the scientific challenges of the EMIL project. This task is organized as follows:

Firstly, we shall evaluate the success of the theory in understanding the development of norm innovation in the Open Source movement by comparing the results of the simulation with the empirical data documented in the Open Source scenarios task. The comparison will lead to a revision and improvement of the theory.

Secondly, the experience obtained from previous tasks will be used to reformulate the theory of norm innovation and to consider its applicability to social phenomena other than those used in its construction and validation. The result from this sub-task will constitute the new model, EMIL-T, as well as a set of further applications demonstrating the generality and the limitations of the theory and models.

Finally, the reformulated theory will be used to re-implement and perhaps modify the models produced in EMIL-S in order to test these models in similar but different cases. We consider this replication and extension to be essential to the demonstration of the validity of our results.

On the methodological side, the practice of evaluating social simulations builds on the long tradition of both computer simulation and experimental research. The methodologies proposed for computational models of complex social phenomena usually require three levels of testing [19, 20, 25]. The first one will have been performed by the simulations execution task (see 7.4)

The second test is for external validity, when the insights and conjectures gained from the model are applied to the real system in laboratory experiments, or in natural conditions [25]. This will be done by the aforementioned comparison between empirical data resulting from Open Sources task (see 7.2).

Finally, the third test is for domain validity. This process involves aligning or docking two (or

more) models that incorporate different mechanisms so as to explain the same phenomena, in order to determine under which conditions they can produce equivalent results [2]. In this project, we will perform an internal test of domain validity, in order to ascertain the minimum agent complexity and the minimum number of agent levels (i.e., individual agents vs. groups, institutions, society) needed to explain the phenomena in object.

## 7. Conclusions

In this paper we have presented the EMIL project, a simulation project on complex social systems, whose objects of investigation are both emergence processes and immergence processes, which are peculiar to complex social systems. As mentioned, the scientific aim of the EMIL project is the analysis of norm innovation by means of a simulation-based investigation.

We have specified a number of crucial questions subtending the whole project (in short, what is the difference, if any, between a convention and a norm? What are the mechanisms by which new norms are conceived; the conditions of their spreading; the circumstances under which they become institutionalised; and the processes through which they decay?), and we have pointed out how the EMIL project shall endeavour to provide an answer to them.

Finally, we have sketched out the EMIL workplan, showing the main scientific and technological challenges, the focal articulations and the links between the tasks.

## 8. Acknowledgements

This work was supported by the EMIL project, funded by the FP6 Information Society and Technologies Programme of the European Commission, in the framework of the FET proactive initiative “Simulating Emergent Properties in Complex Systems”.

The information provided is the sole responsibility of the authors and does not reflect the Community’s opinion. The Community is not responsible for any use that might be made of data appearing in this publication.

## 9. References

[1] R. Axelrod, The Evolution of Co-operation, Basic Books, New York, NY, USA, 1984.

[2] R. Axtell, R. Axelrod, J. M Epstein, and M. D. Cohen, "Replication of Agent-Based Models, Aligning Simulation Models: A Case Study and Results", in R. Axelrod, (Ed.) The Complexity of Cooperation, Princeton University Press, Princeton, NJ, USA, pp. 183-205, 1997.

[3] K. Binmore, Playing Fair: Game Theory and the Social Contract I, Cambridge MA: MIT Press, 1994.

[4] R. Boyd, P. J. Richerson, "Punishment Allows the Evolution of Cooperation (or Anything Else) in Sizable Groups", Ethology & Sociobiology, vol. 13, pp. 171-195, 1992.

[5] C. F. Camerer, E. Fehr, "Measuring social norms and preferences using experimental games: a guide for social scientists", in J.Henrich, S.Bowles, R.Boyd, C.Camerer, E.Fehr, H.Gintis, (Eds), Foundations of Human Sociality, Oxford: Oxford University Press, p. 320, 2004.

[6] C. Castelfranchi, "Emergence and Cognition: Towards a Synthetic Paradigm in AI and Cognitive Science", in H. Coelho (Ed.), Progress in Artificial Intelligence - IBERAMIA 98, LCNS 1484, Springer, pp.13-26, 1998.

[7] C. Castelfranchi, R. Conte, and M. Paolucci, "Normative reputation and the costs of compliance", Journal of Artificial Societies and Social Simulation, vol. 1, no. 3, 1998.

[8] R.H Coase, "The Nature of the Firm: Origin", Journal of Law, Economics and Organization, Oxford University Press, vol. 4(1), pp. 3-17, 1988.

[9] R. Conte, "Memes through (Social) Minds", in R. Auger (Ed), Darwinizing Culture: the Status of Memetics as a Science, Oxford: Oxford University Press, pp. 83-120, 2000.

[10] R. Conte, and C. Castelfranchi, Cognitive and social action, London: London University College of London Press, 1995.

[11] R. Conte, and C. Castelfranchi, "From conventions to prescriptions. Towards a unified theory of norms", AI&Law, vol. 7, pp. 323-340, 1999.

[12] R. Conte, and C. Castelfranchi, "The Mental Path of Norms", Ratio Juris, vol.19 (4),

pp. 501 - 517, 2006

[13] R. Conte, C. Dellarocas, Social order in Multiagent system, Kluwer Academic Publishers, 2001.

[14] R. Conte, B. Edmonds, S. Moss, K. Sawyer, "The Relevance of the Social Sciences for Multi Agent Based Simulation", Computational and Mathematical Organization Theory, vol. 7, pp.183-205, 2001.

[15] V. Dignum, J-J. Meyer, H. Weigand, "Towards an Organizational Model for Agent Societies Using Contracts", in Proc. of AAMAS'02, First International Joint Conference on Autonomous Agents and Multi-agent Systems, Bologna, Italy, July 15 – 19, 2002.

[16] I. Foster, C. Kesselman, S. Tuecke, "The anatomy of the Grid: Enabling scalable virtual organizations", International Journal of High Performance Computing Applications, vol. 15(3), pp. 200-222, 2001.

[17] N. Gilbert, "Emergence in Social Simulation", in N. Gilbert, R. Conte, (Eds), Artificial Societies. A Computer Simulation of Social Life, London: UCL Press, pp. 144-56, 1995.

[18] N. Gilbert, "Open problems in using agent-based models in industrial and labour dynamics", Advances in complex systems, vol. 7(2), pp. 285-288, 2004.

[19] N. Gilbert, and P. Terna, "How to build and use agent-based models in social science", Mind & Society, vol. 1, pp. 55-72, 2000.

[20] N. Gilbert, and K. Troitzsch, Simulation for the social scientist, Milton Keynes: Open University Press, 1999.

[21] O. Hart, Incomplete Contracts and the Theory of the Firm, Working papers 448, Massachusetts Institute of Technology (MIT), Department of Economics, 1987.

[22] D. Kreps, A Course in Microeconomic Theory, Princeton Univ. Press, Princeton, NJ, 1990.

[23] D. Lewis, Convention: A Philosophical Study, Cambridge: Harvard UP, 1969.

[24] G. Lindemann, D. Moldt , and M. Paolucci, (Eds.), Regulated Agent-Based Social Systems, First International Workshop, RASTA 2002, Bologna, Italy, July 16, 2002, Revised Selected

and Invited Papers, LNAI 2934, Springer, 2004.

[25] M. W. Macy and R. Willer, “From factors to actors: computational sociology and agent-based modelling”, Annual Review of Sociology, Vol. 28, pp. 143-166, 2002.

[26] C. W. Reynolds, “Flocks, Herds, and Schools: A Distributed Behavioural Model”, Computer Graphics, vol. 21(4) (SIGGRAPH '87 Conference Proceedings), pp. 25-34, 1987.

[27] T. C. Schelling, The strategy of conflict, Oxford: Oxford University Press, 1960.

[28] T. C Schelling, “Dynamic Models of Segregation”, Journal of Mathematical Sociology, vol. 1, pp. 143-186, 1971.

[29] Y. Shoham and M. Tennenholtz, “On the Synthesis of Useful Social Laws in Artificial Societies”, in Proc. 10th National Conference on Artificial Intelligence, San Mateo, CA, Kaufmann, pp. 276-82, 1992.

[30] R. Sun (Ed.), Cognition and Multi-Agent Interaction: From Cognitive Modeling to Social Simulation, Cambridge: Cambridge University Press, 2005.

[31] K. Troitzsch, “Social Science Simulation - Origins, Prospects, Purposes” in: R. Conte, R. Hegselmann, and P. Terna, (eds.) Simulating Social Phenomena, Berlin (Springer) (LNEMS 456), pp. 41-54, 1997.

[32] G. Weisbuch, Complex systems dynamics and generic properties, EXISTENCE focus document, www.complexityscience.org/NoE/weisbuch.pdf , 2005.

[33] G. Weisbuch, G. Duchateau-Nguyen, Societies, cultures and fisheries from a modeling perspective, Journal of Artificial Societies and Social Simulation, vol. 1(2), 1998.

# Epistemic Norms in a Nutshell

**Emil Weydert**

ILIAS - University of Luxembourg

When modeling agents involved in scientific activities, one of the major issues is how to let them evaluate the reliability of evidence and the trustability of sources. In the real world of science, input judgment is a common, potentially complex task, of considerable interest for scientific practice and meta-scientific studies. It turns out that epistemic norms here constitute an important tool to manage information quality and coherence, as well at the level of individual scientists/groups, as at the level of scientific communities. In fact, the adherence to appropriate rules or prescriptions in the context of research helps to generate trust, and therefore to limit the complexity of information gathering, processing, and communication.

Generally speaking, epistemic norms are just conventions meant to control the epistemic behaviour of cognitive agents in specific contexts. In particular, they are meant to guide the formation and evolution of doxastic attitudes, but also the specification of doxastic goals and intentions. The overarching ideal is of course to gain knowledge, i.e. adequately justified true belief (or so) about the world, as mirrored by the attribute "epistemic". Epistemic norms are often characterized by high expectations w.r.t. some form of "correctness", "optimality", or "rationality". However, in the context of actual resource-bounded agents, these are rather elusive concepts, allowing different interpretations, and hence again different normative stipulations. For practical purposes, it seems therefore preferable to adopt a broad reading of epistemic norms. This more liberal approach forces us however also to take a more dynamic perspective, opening the door to defeasible and revisable epistemic norms. In fact, we even will have to accept the co-existence of – locally and globally – conflicting epistemic norms. The question is now how we are going to model all this. Here we propose two general principles, of course to be supplemented by many others.

**1. Generality.** A theoretical framework for epistemic norms should be applicable to any reasonable model of epistemic communities, whatever its granularity. The minimal ingredients of such a model are evolving decision-taking cognitive agents $a$, reasoning about the concrete/abstract world, communicating with each other, and interacting with the environment $\omega$ (except for math agents).

**2. Normativity.** Whatever the exact nature of epistemic norms, on the semantic level, every admissible set of norms $\mathcal{N}$ should define for each state $\vec{a}$ of an epistemic community the collection $\mathcal{H}_{\vec{a}}[\mathcal{N}]$ of those histories which obey the norms and are compatible with the state. We emphasize that we do not assume $\mathcal{H}_{\vec{a}}[\mathcal{N} \cup \mathcal{N}'] = \mathcal{H}_{\vec{a}}[\mathcal{N}] \cap \mathcal{H}_{\vec{a}}[\mathcal{N}']$. This is necessary, first, to handle conflicting norms, and secondly, to model synergetic ones. However, norms specifying the same choice sets do not have to be identical. The difference may show up through the choice behaviour in the context of other norms.

Dagstuhl Seminar Proceedings 07122
Normative Multi-agent Systems
http://drops.dagstuhl.de/opus/volltexte/2007/924

# Expressing and Verifying Business Contracts with Abductive Logic Programming

Marco Alberti Federico Chesani Marco Gavanelli
Evelina Lamma Paola Mello Marco Montali
Paolo Torroni

February 10, 2007

**Abstract**

In this article, we propose to adopt the $\mathcal{S}$CIFF abductive logic language to specify business contracts, and show how its proof procedures are useful to verify contract execution and fulfilment. $\mathcal{S}$CIFF is a declarative language based on abductive logic programming, which accommodates forward rules, predicate definitions, and constraints over finite domain variables. Its declarative semantics is abductive, and can be related to that of deontic operators; its operational specification is the sound and complete $\mathcal{S}$CIFF proof procedure, defined as a set of transition rules, which has been implemented and integrated into a reasoning and verification tool. A variation of the $\mathcal{S}$CIFF proof-procedure (g-$\mathcal{S}$CIFF) can be used for static verification of contract properties.

We demonstrate the use of the $\mathcal{S}$CIFF language for business contract specification and verification, in a concrete scenario. In order to accommodate integration of $\mathcal{S}$CIFF with architectures for business contract, we also propose an encoding of $\mathcal{S}$CIFF contract rules in RuleML.

## Authors' current affiliations

Marco Alberti, Marco Gavanelli, Evelina Lamma: ENDIF, University of Ferrara, Via Saragat 1, 44100 Ferrara, Italy.

Email: {marco.alberti|marco.gavanelli|evelina.lamma}@unife.it

Federico Chesani, Paola Mello, Marco Montali, Paolo Torroni: DEIS, University of Bologna, Viale del Risorgimento 2, 40123 Bologna, Italy.

Email: {fchesani|pmello|mmontali|ptorroni}@deis.unibo.it

## 1 Introduction

Business contracts are an important conceptual abstraction and a practical guiding and governance mechanism for cross-organizational collaboration. Contracts



Dagstuhl Seminar Proceedings 07122
Normative Multi-agent Systems
http://drops.dagstuhl.de/opus/volltexte/2007/901

can be in fact considered as the main coordination mechanism for the extended enterprise [MGL+04]. A business contract architecture [Mil95] is therefore an important part of the extended enterprise which aims to provide functionalities such as contract management and monitoring. Natural requirements for a contract management framework are a language with clear semantics for contract specification, and operational procedures for (*i*) verifying contract properties at design time, and (*ii*) verifying the conformance of parties to contracts at run time.

From a high-level, functional viewpoint, a contract management system is a component that is fed with the "what" of the problem, by domain expert users, and takes care of the "how," through a suitable execution model. Computational logics offer a broad range of languages and mechanisms that couple declarative ("what is") specification languages with sound operational ("how to") execution models that need not be disclosed to the user of the specification language. For this reason, we strongly believe that computational logic-based frameworks, adequately extended to support event-based monitoring of business activities associated with contracts, should play a key role in contract management systems.

Among the most influential computational logic frameworks for business contract representation and reasoning we find Courteous Logic Programming (CLP, [GLC99]) and Defeasible Logic (DL, [Gov05]), the former being in fact a variant of the latter [AMB00]. These are languages for nonmonotonic reasoning, mainly used in the context of business contracts to enable normative reasoning and to identify and resolve conflicts arising by events and contract rules, reason about violations, specify and enforce reparational obligations, and so on. In this article, in context of contract management systems, we are mailny concerned with the aspect of runtime monitoring and verification of contracts, rather than on the ontological and semantic aspects of contract specification. We focus primarily on the problem of *runtime evaluation* of contract policies, i.e., expressions consisting of behaviour constraints, event patterns and states [MGL+04], to determine whether parties' obligations have been satisfied or whether there are violations to the contract.

We base our work on $\mathcal{S}$CIFF, the computational logic-based language and framework conceived within the context of the SOCS EU project [SOC05] to specify agent interaction protocols. $\mathcal{S}$CIFF consists of a logic language based on abductive logic programming, a sound and complete proof procedure [AGL+05], and a software tool which implements it, based on an efficient inference engine and constraints solving technology [ACG+06b]. First class entities in the $\mathcal{S}$CIFF language are *events*, which represent entities such as actions being taken, timeouts associated with deadlines, and external events such as messages being sent or services being requested, and *expectations*, which describe a desired behaviour in terms of events. Expectations are related with each other and with events by logical epxressions called Integrity Contraints (ICs). ICs express in fact behaviour contraints, and are the main building blocks in the specification of policies. Expectations are modeled in $\mathcal{S}$CIFF as abducible predicates, since they model event that may happen (but we do not know whether that will be the

case). Thus the abductive nature of the framework. Expectations are related to deontic operators such as obligation, prohibition and permission [AGL+06].

In this paper, we propose $\mathcal{S}$CIFF as a language and operational framework to specify and reason upon business contracts. The deontic reading of $\mathcal{S}$CIFF specifications arising from such a relationship is one of the elements that make the $\mathcal{S}$CIFF language, we believe, a good candidate as a contract specification and reasoning language. Reasoning upon contract specifications (and events) can be done at two different stages of contract design and enactment: at run-time, for example in the way that we propose, and at design time, as it is the case with DL and CLP. We consider it important to enable these two kinds of verification within the same framework and, if possible, using the same specification language, in order to minimize translation errors and the unavoidable unaccuracy caused by different languages. To this end, an extension of $\mathcal{S}$CIFF, called g-$\mathcal{S}$CIFF, has been defined to verify protocol properties at design time [ACG+06c], and we show here how it can be used to enable design-time reasoning on contracts.

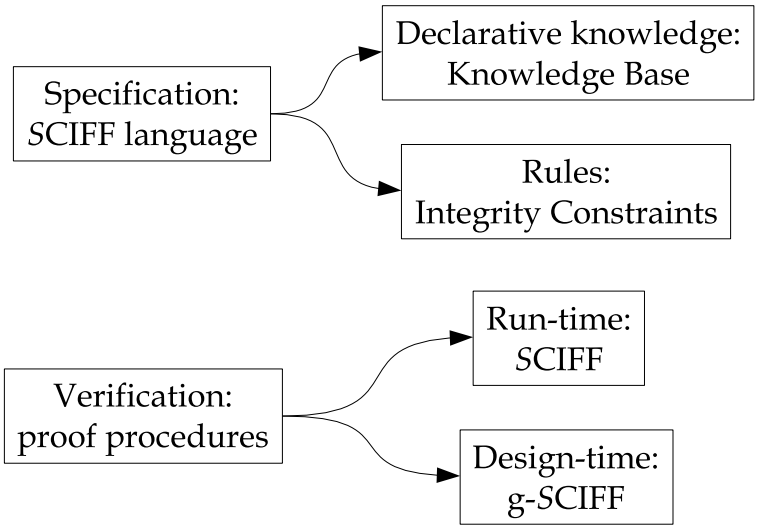



Figure 1: Contract specification and verification in the $\mathcal{S}$CIFF framework.

Fig. 1 summarizes the components of the $\mathcal{S}$CIFF framework that can be used in contract specification and verification.

The paper is structured as follows. Sect. 2 is devoted to the specification of contract in the $\mathcal{S}$CIFF language: we first give the necessary background, by presenting the syntax (Sect. 2.1) and declarative semantics (Sect. 2.2) of the $\mathcal{S}$CIFF language, then we show how $\mathcal{S}$CIFF expectations are related to deontic operators (Sect. 2.3) and finally we demonstrate $\mathcal{S}$CIFF in a concrete scenario, by proposing a possible specification of contract clauses (Sect. 2.4). In Sect. 3, we investigate the verification issues: first run-time verification (Sect. 3.1), and

then design-time verification (Sect. 3.2), recalling, in each case, the relevant proof procedure ($\mathcal{S}$CIFF and g-$\mathcal{S}$CIFF, respectively).

The integration of $\mathcal{S}$CIFF with architectures for business contract is facilitated by a suitable encoding of $\mathcal{S}$CIFF contract rules in RuleML, summarized in Sect. 4. A discussion of related work follows.

# 2 Contract specification

A contract in the $\mathcal{S}$CIFF language is basically specified by means of two components: a knowledge base, which defines declaratively domain-specific knowledge (such as deadlines) and a set of integrity constraints, which describe contract clauses and can be seen as forward rules that generate expectations about the behaviour of the parties involved in the contract. A declarative semantics based on abductive logic programming determines whether the parties have complied to the contract. A useful feature of the $\mathcal{S}$CIFF language is its integration of Constraint Logic Programming [JM94], which makes deadlines easy to specify and efficient to verify.

## 2.1 Syntax of the $\mathcal{S}$CIFF language

The $\mathcal{S}$CIFF language is composed of entities for expressing:

- events and expectations about events;
- relationships between events and expectations.

### 2.1.1 Representation of the behaviour of parties

**Events** *Events* are the abstractions used to represent the actual behaviour.

**Definition 2.1.** *An* event *is an atom:*

- *with predicate symbol* **H***;*
- *whose first argument is a ground term; and*
- *whose second argument is an integer.*

Intuitively, the first argument is meant to represent the description of the happened event, according to application-specific conventions, and the second argument is meant to represent the time at which the event has happened.

In this paper, we map all events to communicative events, identified by the functor *tell*. In particular, the description of happened events is of the format

$$tell(Sender, Receiver, Content[, Dialog]),$$

where the optional *Dialog* parameter is an identifier of the interaction being described and the other arguments have the obvious meaning.

**Example 2.2.**

$$\mathbf{H}(\ tell(\ telco,\ c,\ phone\_bill(390512093086, 145886, 205),\ 19). \qquad (1)$$

*says that telco sent to c a phone_bill (for the phone number 390512093086, whose identifier is 145886 and whose amount is 205) at time 19.*

A *negated event* is a negative literal **not H**(...,...). We will call *history* a set of happened events, and denote it with the symbol **HAP**.

**Expectations** *Expectations* are the abstractions used to represent the desired events from an external viewpoint. They represent the ideal behaviour of the system, i.e., the actions that, once performed, would make the system compliant to its specifications. Our choice of the terminology "expectation" is intended to stress that events cannot be enforced, but only expected, to be as we would like them to be.

Expectations are of two types:

- *positive*: representing some event that is expected to happen;
- *negative*: representing some event that is expected *not* to happen.

**Definition 2.3.** *A* positive expectation *is an atom:*

- *with predicate symbol* **E***;*
- *whose first argument is a term; and*
- *whose second argument is a variable or an integer.*

Intuitively, the first argument is meant to represent an event description, and the second argument is meant to tell for what time the event is expected (which should not be confused with the time at which the expectation is generated, which is not modeled by $\mathcal{S}$CIFF's declarative semantics). Expectations may contain variables, which leaves the expected event not completely specified. Variables in positive expectations are always existentially quantified: if the time argument is a variable, for example, this means that the event is expected to happen at *any* time. We do not associate a specific semantics to time; we rather treat an expectation's time argument as any other variable. This choice simplifies the $\mathcal{S}$CIFF language's declarative and operational semantics.

**Example 2.4.** *The atom*

$$\mathbf{E}(\ tell(\ telco,\ c,\ phone\_bill(390512093086, Id, Amount), T). \qquad (2)$$

*says that telco is expected to send to c a phone_bill (for the number 390512093086, whose identifier is Id and whose amount is Amount) at time T.*

A *negated positive expectation* is a positive expectation with the explicit negation operator ¬ applied to it. As explained in Sect. 2.1.2, variables in negated positive expectations are quantified as those in positive expectations.

**Definition 2.5.** *A* negative expectation *is an atom:*

- *with predicate symbol* **EN***;*
- *whose first argument is a term; and*
- *whose second argument is a variable or an integer.*

Intuitively, the first argument is meant to represent an event description, and the second argument is meant to tell in which time points the event is expected not to happen. As well as positive expectations, negative expectations may contain variables, which are typically universally quantified[1]: for example, if the time argument is a variable, then the event is expected not to happen at *all* times.

**Example 2.6.** *The atom*

$$\mathbf{E}(\ tell(\ telco,\ c,\ phone\_bill(390512093086, Id, Amount), T). \tag{3}$$

*means says that telco is expected* not *to send to c a phone_bill (for the number 390512093086, with any Id and for any Amount) at any time T.*

A *negated negative expectation* is a negative expectation with the explicit negation operator $\neg$ applied to it. As explained in Sect. 2.1.2, variables in negated negative expectations are quantified as those in negative expectations.

Note that $\neg\mathbf{E}(tell(bob, alice, refuse(phone\_number), dialog\_id), T_r)$ is different from $\mathbf{EN}(tell(bob, alice, refuse(phone\_number), dialog\_id), T_r)$. The intuitive meaning of the former is: no *refuse* is expected by Bob (if he does, we simply did not expect him to), whereas the latter has a different, stronger meaning: it is expected that Bob does not utter *refuse* (by doing so, he would frustrate our expectations).

The syntax of events and expectations is summarised in Tab. 2.1, and it will be used as such by the subsequent Tab. 2.2 and 2.3. The syntactical entities *ExistLiteral* and *NbfLiteral* will also be used in the subsequent Tab. 2.2 and 2.3.

### 2.1.2 Contract specifications

A contract specification, i.e, a specification of the interaction in the $\mathcal{S}$CIFF framework, is composed of two elements:

- A *Knowledge Base*;
- A set of *Integrity Constraints*.

[1]For a complete treatment of quantification in the $\mathcal{S}$CIFF language, we refer the interested reader to [ACG+06d].

**Table 2.1** Syntax of events and expectations

| | | |
|---|---|---|
| *EventLiteral* | ::= | [**not**]*Event* |
| *Event* | ::= | **H**(*GroundTerm*, *Integer*) |
| *ExpLiteral* | ::= | *PosExpLiteral* \| *NegExpLiteral* |
| *PosExpLiteral* | ::= | [¬]*PosExp* |
| *NegExpLiteral* | ::= | [¬]*NegExp* |
| *PosExp* | ::= | **E**( *Term*, *Variable* \| *Integer*) |
| *NegExp* | ::= | **EN**( *Term*, *Variable* \| *Integer*) |
| *ExistLiteral* | ::= | *PosExpLiteral* \| *Literal* |
| *NbfLiteral* | ::= | **not** *Atom* |
| *Literal* | ::= | [**not**]*Atom* |

**Knowledge Base** The Knowledge Base ($KB_S$) is a set of *Clause*s in which the body can contain (besides defined literals) expectation literals and restrictions. [2]

Intuitively, the $KB_S$ is used to express declarative knowledge about the specific application domain.

**Table 2.2** Syntax of the Knowledge Base

| | | |
|---|---|---|
| $KB_S$ | ::= | $[Clause]^\star$ |
| *Clause* | ::= | $Head \leftarrow Body$ |
| *Head* | ::= | *Atom* |
| *Body* | ::= | $ExtLiteral\ [\ \wedge\ ExtLiteral\ ]^\star\ [:\ Restriction\ [,\ Restriction\ ]^\star\ ]\ |\ true$ |
| *ExtLiteral* | ::= | *Literal* \| *ExpLiteral* |

The syntax of the Knowledge Base is given in Tab. 2.2, and it will be used as such also in Tab. 2.3.

**Goal** In the $\mathcal{S}$CIFF framework, the role of the *goal* is the same as in the logic programming literature, i.e., a predicate that should be entailed. Therefore, the term "goal" does not necessarily have the typical connotation (of "common" or "social" goal) found in multi-agent systems literature, though it can be used for such a purpose.

The syntax of the goal is the same as the *Body* of a clause (Tab. 2.2). The quantification rules are the following:

[2] In the $\mathcal{S}$CIFF language, restrictions can be considered as CLP constraints [JM94], that can also be applied to universally quantified variables with the semantics defined by Bürckert [Bür94].

- All variables that occur in an *ExistLiteral* are existentially quantified.
- All remaining variables are universally quantified.

**Integrity Constraints** Integrity Constraints (also ICs, for short, in the following) are implications that, operationally, are used as forward rules, as will be explained in Sect. 3. Declaratively, they relate the various entities in the $\mathcal{S}$CIFF framework, i.e., expectations, events, and constraints/restrictions, together with the predicates in the knowledge base.

**Table 2.3** Syntax of Integrity Constraints (ICs)

| | | |
|---|---|---|
| $\mathcal{IC}_S$ | ::= | $[IC]^\star$ |
| $IC$ | ::= | $Body \rightarrow Head$ |
| $Body$ | ::= | $(EventLiteral \mid ExpLiteral)\ [\ \wedge BodyLiteral\ ]^\star$ $[: Restriction\ [,\ Restriction\ ]^\star\ ]$ |
| $BodyLiteral$ | ::= | $EventLiteral \mid ExtLiteral$ |
| $Head$ | ::= | $HeadDisjunct\ [\ \vee HeadDisjunct\ ]^\star \mid false$ |
| $HeadDisjunct$ | ::= | $HeadLiteral\ [\ \wedge HeadLiteral]^\star\ [: Restriction\ [,\ Restriction\ ]^\star\ ]$ |
| $HeadLiteral$ | ::= | $Literal \mid ExpLiteral$ |

The syntax of ICs is given in Tab. 2.3: the *Body* of ICs can contain conjunctions of all elements in the language (namely, **H**, **E**, and **EN** literals, defined literals and restrictions), and their *Head* contains a disjunction of conjunctions of any of the literals in the language, except for **H** literals.

**Contract Specification** Given a Knowledge Base $KB_S$ and a set $\mathcal{IC}_S$ of Integrity Constraints, we call the pair $\langle KB_S, \mathcal{IC}_S \rangle$ a *Contract Specification*. Intuitively, a contract specification is a description of the acceptable, or desirable, histories, as defined by its declarative semantics, given formally in Sect. 2.2.

## 2.2 Declarative Semantics

In the following, we describe the (abductive) declarative semantics of the $\mathcal{S}$CIFF framework, which is inspired by other abductive frameworks such as the IFF by Fung and Kowalski [FK97], but introduces the concept of fulfilment, used to express a correspondence between the expected and the actual events. The declarative semantics of a contract specification is given for each specific history (see Sect. 2.1.1). We call a specification grounded on a history an *instance* of the contract.

**Definition 2.7. Contract instance** *Given a contract specification* $\mathcal{S} = \langle KB_S, \mathcal{IC}_S \rangle$ *and a history* **HAP**, $\mathcal{S}_{\mathbf{HAP}}$ *represents the pair* $\langle \mathcal{S}, \mathbf{HAP} \rangle$, *called the* **HAP**-instance of $\mathcal{S}$ *(or simply an* instance of $\mathcal{S}$*).*

In this way, $\mathcal{S}_{\mathbf{HAP}^i}$, $\mathcal{S}_{\mathbf{HAP}^f}$ will denote different instances of the same contract specification $\mathcal{S}$, based on two different histories: $\mathbf{HAP}^i$ and $\mathbf{HAP}^f$.

We adopt an abductive semantics for the contract instance. Declaratively, a ground set **EXP** of hypotheses should entail the goal and satisfy the integrity constraints. In our case the set **EXP** of hypotheses is, in particular, a set of ground expectations, positive and negative, possibly negated by explicit negation. Notice that, by virtue of explicit negation, all of such expectations are positive abducible literals in ALP terminology.

**Definition 2.8. Abductive explanation** *Given a contract specification* $\mathcal{S} = \langle KB_S, \mathcal{IC}_S \rangle$, *an instance* $\mathcal{S}_{\mathbf{HAP}}$ *of* $\mathcal{S}$, *and a goal* $\mathcal{G}$, **EXP** *is an abductive explanation of* $\mathcal{S}_{\mathbf{HAP}}$ *for goal* $\mathcal{G}$ *if:*

$$Comp(KB_S \cup \mathbf{HAP} \cup \mathbf{EXP}) \cup \text{CET} \cup T_{\mathcal{X}} \models \mathcal{IC}_S \quad (4)$$

$$Comp(KB_S \cup \mathbf{EXP}) \cup \text{CET} \cup T_{\mathcal{X}} \models \mathcal{G} \quad (5)$$

*where Comp represents the three-valued* completion *of a theory [Kun87],* CET *is Clark [Cla78] Equational Theory, and* $T_{\mathcal{X}}$ *is the constraint theory [JM94].*

The symbol $\models$ is interpreted in three valued logics. In particular, if we interpret expectations as abducible predicates, we can rely upon a three-valued model-theoretic semantics as intended meaning, as done, for instance, in a different context, by Fung and Kowalski [FK97], Denecker and De Schreye [DS98].

We also require consistency with respect to explicit negation [AB94] and between positive and negative expectations.

**Definition 2.9. ¬-consistency** *A set* **EXP** *of expectations is* ¬-consistent *if and only if for each (ground) term* $p$ *and integer* $t$*:*

$$\{\mathbf{E}(p,t), \neg\mathbf{E}(p,t)\} \not\subseteq \mathbf{EXP} \qquad \textit{and} \qquad \{\mathbf{EN}(p,t), \neg\mathbf{EN}(p,t)\} \not\subseteq \mathbf{EXP}. \quad (6)$$

**Definition 2.10. E-consistency** *A set* **EXP** *of expectations is* **E**-consistent *if and only if for each (ground) term* $p$ *and integer* $t$*:*

$$\{\mathbf{E}(p,t), \mathbf{EN}(p,t)\} \not\subseteq \mathbf{EXP} \quad (7)$$

The following definition establishes a link between happened events and expectations, by requiring positive expectations to be matched by events, and negative expectations not to be matched by events.

**Definition 2.11. Fulfillment** *Given a history* **HAP**, *a set* **EXP** *of expectations is* **HAP**-fulfilled *if and only if*

$$\forall \mathbf{E}(p,t) \in \mathbf{EXP} \Rightarrow \exists \mathbf{H}(p,t) \in \mathbf{HAP} \;\; \textit{and} \;\; \forall \mathbf{EN}(p,t) \in \mathbf{EXP} \Rightarrow \nexists \mathbf{H}(p,t) \in \mathbf{HAP} \quad (8)$$

*Otherwise,* **EXP** *is* **HAP**-violated.

When all the given conditions (4-8) are met for at least one set of expectations **EXP**, we say that the goal is *achieved* and **HAP** is compliant to $\mathcal{S}$ with respect to $\mathcal{G}$ and **EXP**, and we write $\mathcal{S}_{\mathbf{HAP}} \models_{\mathbf{EXP}} \mathcal{G}$. In particular:

| Operator | Abducibile |
|---|---|
| **Forb** $A$ | $\mathbf{EN}(A)$ |
| **Obl** $A$ | $\mathbf{E}(A)$ |
| **Perm** $A$ | $\neg\mathbf{EN}(A)$ |
| **Perm** $NONA$ | $\neg\mathbf{E}(A)$ |

Table 1: Deontic notions as expectations

**Definition 2.12. Goal achievement** *Given an instance* $\mathcal{S}_{\mathbf{HAP}}$ *of a contract specification* $\mathcal{S} = \langle KB_S, \mathcal{IC}_S \rangle$ *and a goal* $\mathcal{G}$*, iff there exists an* **EXP** *that is an abductive explanation of* $\mathcal{S}_{\mathbf{HAP}}$ *for* $\mathcal{G}$*, and it is* $\neg$-consistent*,* **E**-consistent *and* **HAP**-fulfilled*, we say that* $\mathcal{G}$ *is* achieved w.r.t. **EXP** *(and we write* $\mathcal{S}_{\mathbf{HAP}} \models_{\mathbf{EXP}} \mathcal{G}$*). Given an instance* $\mathcal{S}_{\mathbf{HAP}}$ *and a goal* $\mathcal{G}$*, we say that* $\mathcal{G}$ *is* achieved *if* $\exists$**EXP** *such that* $\mathcal{G}$ *is achieved w.r.t.* **EXP***.*

In the remainder of this article, when we simply say that a history **HAP** is compliant to a contract specification $\mathcal{S}$, we will mean that **HAP** is compliant to $\mathcal{S}$ with respect to the goal *true*. We will say that a history **HAP** *violates* a specification $\mathcal{S}$ to mean that **HAP** is not compliant to $\mathcal{S}$. When **HAP** is apparent from the context, we will often omit mentioning it.

## 2.3 Expectations and deontic operators

In this section, we recall the mapping from deontic operators (obligation, permission, prohibition) to the expectations of the $\mathcal{S}$CIFF framework, proposed in [AGL$^+$06].

Such a mapping can be used to attribute a deontic meaning to $\mathcal{S}$CIFF-based contract specifications.

The mapping is shown in Tab. 1.

The first line of the table proposes a correspondence between the notion of prohibition (which requires an action not to be performed) and ours of negative expectation (which requires an event not to belong to the history).

In fact, the correspondence is more apparent looking at Def. 2.11, which requires, for a set of expectation to be fulfilled, the absence, in the history of events, of any event matching a negative expectation. This definition resembles closely the reduction of the prohibition operator proposed by [Mey88], where "it is forbidden to perform (an action) $\alpha$ in (a state) $\sigma$ iff one performs $\alpha$ in $\sigma$ one gets into trouble" (in that paper, "trouble" means an "undesirable state of affairs"; which is a goode description of our state of violation).

Reasoning in a similar way, it is possible to notice a correspondence between the notion of obligation (which requires an action to be performed) and ours of positive expectation (which requires an event to belong to the history), as shown in the second line in Tab. 1.

Moreover, since a negative expectation $\mathbf{EN}(A)$ has to be read as *it is expected not* $A$ (i.e., it is a shorthand for $\mathbf{E}(\textit{not } A)$), its (explicit) negation, $\neg\mathbf{EN}(A)$, corresponds to permission of $A$. Finally, due to the logical relations among

obligation, prohibition and permission discussed in [Sar04], the fourth line of Table 1 shows how to map permission of a negative action.

[AGL$^+$06] provides a formal support of this mapping, based on a correspondence between the Kripke semantics of deontic operators and the declarative semantics of the $\mathcal{S}$CIFF frameworks.

## 2.4 Sample contract specification

In this section, we intend to demonstrate how to specify, inside the $\mathcal{S}$CIFF framework, contracts that may result too intricate for a representation based on other formalisms, such as finite state machines, coloured Petri nets, or AUML diagrams. The example we give is a simplified version of a real life situation, describing the activation of a telephone line (carrier) by a customer. We consider the clauses of the contract a user must sign as the building blocks of a contract, which makes use of expressive combinations of **E**, **EN**, and **H** predicates, CLP constraints and predicates defined in the $KB_S$. With $\mathcal{S}$CIFF we give a faithful representation of such a contract, which makes it understandable, modular, and verifiable. Despite all effort put by the telephone company into making things as obscure as possible, at any time we (as customers) will be able to detect, via $\mathcal{S}$CIFF, whether the telephone company (*telco* in the following) has the right to interrupt the service or to request a payment from us, and whether we have the right to complain with *telco*, and not to pay part of the bill. Similarly, *telco* will receive indications about when to send requests for payment, or when (not) to activate or (not) to de-activate the carrier.

### 2.4.1 Description of the contract

The procedures that regulate the concession of a carrier to a customer are contained in a contract, that the parties (*telco* and the customer) agree upon. The contract is composed of several parts, stating what to do when the customer request a new carrier, the procedures for paying the bills, for handling complaints, what obligations/penalties apply in case of late payments, and how to delegate authority to the relevant bureaus, to make any necessary determination as to whether the parties have complied with all requirements as set forth in the contract. We enucleated a set of clauses in the contract, and gave a specifications of them in the $\mathcal{S}$CIFF framework. ICs are reported in Spec. 2.1, and the $KB_S$ is reported in Spec. 2.2. We chose a set of clauses about bill and complaint handling:

1. After sending a phone bill to a customer, *telco* cannot send requests for payment before a pre-defined amount of time (call it $TWait$ has passed;
2. after $TWait$, either the customer has paid for the bill, or filed a complaint, or *telco* is allowed to send a request for payment;
3. after receiving a legitimate request for payment, either the customer pays for the bill, or *telco* is allowed to de-activate the carrier, after a further $TWait$;

4. if, upon receiving a request for payment, the customer pays by *TWait*, *telco* is not allowed to de-activate the carrier;

5. if a customer files, by *TWait*, an admissible complaint about a received bill, the customer is no longer expected to pay for it, and *telco* is not allowed to request a payment.

### 2.4.2 $\mathcal{S}$CIFF specification of the contract

Spec. 2.1 contains five ICs: roughly speaking, the first three describe in general what is the expected behaviour of *telco*, regarding bill handling, whereas the last two are about the rights of the customer (*C*). The ICs state the following:

- by $[IC1]$, after sending a bill at time *T1*, *telco* may not send requests for payments before time $T1 + TWait$, where *TWait* is the amount of time defined by the *default_wait* predicate in the $KB_S$.

- by $[IC2]$, after *telco* sends a bill at time *T1*, one of the following expectations hold: either *C* pays the bill in full by $T1 + TWait$, or *C* complains about (part of) the bill by $T1 + TWait$, or *telco* gains the right to send a request or payment at some time *T4* later than $T1 + TWait$ . We shall notice that all complaints that *C* possibly sends after the deadline ($T1 + TWait$) will not have an impact on the state of affairs in these procedures, since they will not match with any expectation;

- by $[IC3]$, if *telco* sent a bill, and later a request for payment at a time in which it was not expected not to do so, and if the request for payment concerns the bill in full, then ether *C* pays the bill, or *telco* gains the right to de-activate the carrier (although *telco* is not obliged to do so);

- by $[IC4]$, if *C* has paid the bill by the deadline, then *telco* cannot de-activate the carrier. Notice that $[IC4]$ fires independently of *telco* actually having the right to send a request for payments;

- by $[IC5]$, after *C* complains about some part of the bill (*Partl_Amnt*), he is no longer expected to pay the bill *Bill_Amnt*.

In the $KB_S$ part of the $\mathcal{S}$CIFF program, shown in Specs. 2.2, we specify deadlines, as in the previous example, and we define what an "admissible complaint" is. To this end, we define a predicate *is_admissible_complaint/2*, which relies upon a database of bills ("list of bills"). In this simplified example, the database is mimicked by a predicate named *list_of_bills/1*. The predicate *member/2* used by *is_admissible_complaint/2* is predefined in most Prolog distributions; this example in particular uses the implementation that comes together with [SIC06].

**Specification 2.1** $\mathcal{IC}_S$ in the contract between *telco* ($T$) and a customer ($C$).

$[IC1]$ $\mathbf{H}(tell(T, C, phone\_bill(Phone\_No, Bill\_Id, Bill\_Amnt), D), T1) \wedge$
$default\_wait(TWait)$
$\rightarrow$ $\mathbf{EN}(tell(T, C, request\_payment(Phone\_No, Bill\_Id, Any\_Amnt), D), T2)$,
$T2 > T1, \quad T2 < T1 + TWait.$

$[IC2]$ $\mathbf{H}(tell(T, C, phone\_bill(Phone\_No, Bill\_Id, Bill\_Amnt), D), T1) \wedge$
$default\_wait(TWait)$
$\rightarrow$ $\mathbf{E}(tell(C, T, pay(Phone\_No, Bill\_Id, Bill\_Amnt, Paymt\_Rcpt), D), T2)$,
$T2 < T1 + TWait$
$\vee$ $\mathbf{E}(tell(C, T, complain(Phone\_No, Bill\_Id, Partl\_Amnt), D), T3)$,
$T3 < T1 + TWait$
$\vee$ $\neg\mathbf{EN}(tell(T, C, request\_payment(Phone\_No, Bill\_Id, Bill\_Amnt), D), T4)$,
$T4 > T1 + TWait.$

$[IC3]$ $\mathbf{H}(tell(T, C, phone\_bill(Phone\_No, Bill\_Id, Bill\_Amnt), D), T1) \wedge$
$\mathbf{H}(tell(T, C, request\_payment(Phone\_No, Bill\_Id, Bill\_Amnt), D), T2) \wedge$
$\neg\mathbf{EN}(tell(T, C, request\_payment(Phone\_No, Bill\_Id, Bill\_Amnt), D), T2) \wedge$
$default\_wait(TWait)$
$\rightarrow$ $\neg\mathbf{EN}(tell(T, C, de\_activate(Phone\_No, reason(Bill\_Id)), D), T3)$,
$T3 > T2 + TWait$
$\vee$ $\mathbf{E}(tell(C, T, pay(Phone\_No, Bill\_Id, Bill\_Amnt, Paymt\_Rcpt), D), T4)$,
$T4 < T2 + TWait.$

$[IC4]$ $\mathbf{H}(tell(T, C, request\_payment(Phone\_No, Bill\_Id, Bill\_Amnt), D), T1) \wedge$
$\mathbf{H}(tell(C, T, pay(Phone\_No, Bill\_Id, Bill\_Amnt, Paymt\_Rcpt), D), T2) \wedge$
$default\_wait(TWait) \wedge T2 < T1 + TWait$
$\rightarrow$ $\mathbf{EN}(tell(T, C, de\_activate(Phone\_No, reason(Bill\_Id)), D), T3).$

$[IC5]$ $\mathbf{H}(tell(T, C, phone\_bill(Phone\_No, Bill\_Id, Bill\_Amnt), D), T1) \wedge$
$\mathbf{H}(tell(C, T, complain(Phone\_No, Bill\_Id, Partl\_Amnt), D), T2) \wedge$
$default\_wait(TWait) \wedge T2 < T1 + TWait \wedge$
$is\_admissible\_complaint(Bill\_Id, Partl\_Amnt)$
$\rightarrow$ $\neg\mathbf{E}(tell(C, T, pay(Phone\_No, Bill\_Id, Partl\_Amnt, Paymt\_Rcpt), D), T3)$,
$T3 > T1$,
$\mathbf{EN}(tell(T, C, request\_payment(Phone\_No, Bill\_Id, Bill\_Amnt), D), T4).$

**Specification 2.2** $KB_S$ in the contract between *telco* and a customer.

$KB_S$ :

$society\_goal.$

$default\_wait(10).$

$is\_admissible\_complaint(Bill\_Id, Partl\_Amnt) \leftarrow$
$list\_of\_bills(L1),$
$member((Bill\_Id, Total\_Amnt), L1),$
$Partl\_Amnt < Total\_Amnt.$

$list\_of\_bills([(145886, 205), (114477, 407), (168945, 126)]).$

# 3 Contract verification

In the following, we describe two types of verification supported by the $\mathcal{S}$CIFF framework: in Sect. 3.1, a verification that the parties involved in a contract are interacting according to it, and in Sect. 3.2, a formal verification of whether a contract enjoys some properties.

## 3.1 Run-time verification

The run-time verification of contracts specified the $\mathcal{S}$CIFF language is performed by means of an abductive proof procedure, called itself $\mathcal{S}$CIFF [AGL$^+$05]. We first recall the $\mathcal{S}$CIFF proof procedure, and then show its behaviour on samples interactions regulated by the contract described in Sect. 2.4.

### 3.1.1 The $\mathcal{S}$CIFF proof procedure

Since the $\mathcal{S}$CIFF language and its declarative semantics are closely related with those of the IFF abductive framework [FK97], the $\mathcal{S}$CIFF proof procedure has also been inspired by the IFF proof procedure. $\mathcal{S}$CIFF is a substantial extension of IFF, and the main differences between the frameworks are, in a nutshell:

- $\mathcal{S}$CIFF supports the dynamical happening of events, i.e., the insertion of new facts in the knowledge base during the computation;
- $\mathcal{S}$CIFF supports universally quantified variables in abducibles;
- $\mathcal{S}$CIFF supports quantifier restrictions;
- $\mathcal{S}$CIFF supports the concepts of fulfilment and violation (see Def. 2.11).

The $\mathcal{S}$CIFF proof procedure is based on a rewriting system transforming one node to another (or to others). In this way, starting from an initial node,

it defines a proof tree. A node can be either the special node *false*, or defined by the tuple

$$T \equiv \langle R, CS, PSIC, \mathbf{PEND}, \mathbf{HAP}, \mathbf{FULF}, \mathbf{VIOL} \rangle. \tag{9}$$

We partition the set of expectations **EXP** into the fulfilled (**FULF**), violated (**VIOL**), and pending (**PEND**) expectations. The other elements are:

- $R$ is the resolvent: a conjunction, whose conjuncts can be literals or disjunctions of conjunctions of literals;
- $CS$ is the constraint store: it contains CLP constraints and quantifier restrictions;
- $PSIC$ is a set of implications, called partially solved integrity constraints
- **HAP** is the history of happened events, represented by a set of events, plus a *closed*(**HAP**) boolean attribute.

If one of the elements of the tuple is *false*, then the tuple is the special node *false*, without successors.

**Initial Node and Success** A derivation $D$ is a sequence of nodes

$$T_0 \rightarrow T_1 \rightarrow \cdots \rightarrow T_{n-1} \rightarrow T_n.$$

Given a goal $\mathcal{G}$, a set of integrity constraints $\mathcal{IC}_S$, and an initial history $\mathbf{HAP}^i$, we build the first node in the following way:

$$T_0 \equiv \langle \{\mathcal{G}\}, \emptyset, \mathcal{IC}_S, \emptyset, \mathbf{HAP}^i, \emptyset, \emptyset \rangle,$$

with *closed(*$\mathbf{HAP}^i$*) = false*. The other nodes are obtained by applying the transitions defined in the next section, until no further transition can be applied.

**Definition 3.1.** *Given an instance* $\mathcal{S}_{\mathbf{HAP}^i}$ *of a contract specification* $\mathcal{S} = \langle KB_S, \mathcal{IC}_S \rangle$ *and a set* $\mathbf{HAP}^f \supseteq \mathbf{HAP}^i$ *there exists a* successful derivation *for a goal* $G$ *iff the proof tree with root node* $\langle \{G\}, \emptyset, \mathcal{IC}_S, \emptyset, \mathbf{HAP}^i, \emptyset, \emptyset \rangle$ *has at least one leaf node*

$$\langle \emptyset, CS, PSIC, \mathbf{PEND}, \mathbf{HAP}^f, \mathbf{FULF}, \emptyset \rangle$$

*where* $CS$ *is consistent, and* **PEND** *contains only negations of expectations* $\neg\mathbf{E}$ *and* $\neg\mathbf{EN}$*. In such a case, we write:*

$$\mathcal{S}_{\mathbf{HAP}^i} \vdash^{\mathbf{HAP}^f}_{\mathbf{EXP}} \mathcal{G}.$$

From a non-failure leaf node $N$, answers (called *expectation answers*) can be extracted in a similar way to the IFF proof procedure. To compute an expectation answer, a substitution $\sigma'$ is computed such that

- $\sigma'$ replaces all variables in $N$ that are not universally quantified by a ground term

- $\sigma'$ satisfies all the constraints in the store $CS_N$.

If the constraint solver is (theory) complete [JM94] (i.e., for each set of constraints $c$, the solver always returns *true* or *false*, and never *unknown*), then there will always exist a substitution $\sigma'$ for each non-failure leaf node $N$. If the solver is incomplete, $\sigma'$ may not exist. The non-existence of $\sigma'$ is discovered during the answer extraction phase. In such a case, the node $N$ will be marked as a failure node, and another non-failure node can be selected (if there is one).

**Definition 3.2.** *Let $\sigma = \sigma'|_{vars(G)}$ be the restriction of $\sigma'$ to the variables occurring in the initial goal $\mathcal{G}$. Let $\Delta_N = (\mathbf{FULF}_N \cup \mathbf{PEND}_N)\sigma'$. The pair $(\Delta_N, \sigma)$ is the* expectation answer *obtained from the node $N$.*

### 3.1.2 $\mathcal{S}$CIFF properties

In the following, we state the most significant formal properties of the $\mathcal{S}$CIFF proof procedure. For the proofs, the interested reader can refer to [ACG+06d].

**Termination** Termination is proven, as for SLD resolution [AB91], for *acyclic* knowledge bases and *bounded* goals and implications. The notion of acyclicity of an abductive logic program is an extension of the corresponding notion given for SLD resolution. Intuitively, for SLD resolution a level mapping must be defined, such that the head of each clause has a higher level than the body. For the IFF, since it contains integrity constraints that are propagated forward, the level mapping should also map atoms in the body of an IC to higher levels than the atoms in the head; moreover, this should also hold considering possible unfoldings of literals in the body of an IC [Xan03]. Similar considerations hold also for $\mathcal{S}$CIFF. We extended the level mapping for considering also CLP constraints. For definitions of boundedness and acyclicity for the contract specification, the reader can refer to [Xan03].

**Theorem 3.3** (Termination of $\mathcal{S}$CIFF)**.** *Let $\mathcal{G}$ be a query to a contract $\mathcal{S} = \langle KB_S, \mathcal{IC}_S \rangle$, where $KB_S$, $\mathcal{IC}_S$ and $\mathcal{G}$ are acyclic w.r.t. some level mapping, and $\mathcal{G}$ and all implications in $\mathcal{IC}_S$ are bounded w.r.t. the level-mapping. Then, every $\mathcal{S}$CIFF derivation for $\mathcal{G}$ for each instance of $\mathcal{G}$ is finite, assuming that happening is not applied.*

*Moreover, under the following conditions:*

- *the number of happened events is finite,*
- *happening is applied only when no other transitions can be applied, and*
- *non-happening has higher priority than other transitions,*

*$\mathcal{S}$CIFF terminates also with dynamically incoming events.*

**Soundness** The $\mathcal{S}$CIFF proof-procedure uses a constraint solver, so its soundness depends on the solver. We proved soundness for a limited solver, containing only the rules for equality and disequality of terms.

**Theorem 3.4** (Soundness of $\mathcal{S}$CIFF)**.** *Given a contract instance* $\mathcal{S}_{\mathbf{HAP}^f}$*, if*

$$\mathcal{S}_{\mathbf{HAP}^i} \vdash^{\mathbf{HAP}^f}_{\mathbf{EXP}} \mathcal{G}$$

*for some* $\mathbf{HAP}^i \subseteq \mathbf{HAP}^f$*, with expectation answer* $(\mathbf{EXP}, \sigma)$*, then*

$$\mathcal{S}_{\mathbf{HAP}^f} \models_{\mathbf{EXP}\sigma} \mathcal{G}\sigma$$

**Completeness** Completeness states that if goal $G$ is achieved under the expectation set $\mathbf{EXP}$, then a successful derivation can be obtained for $G$, possibly computing a set $\mathbf{EXP}'$ of the expectations whose grounding (according to the expectation answer) is a subset of $\mathbf{EXP}$.

**Theorem 3.5.** *Given a contract instance* $\mathcal{S}_{\mathbf{HAP}}$*, a (ground) goal* $G$*, for any ground set* $\mathbf{EXP}$ *such that* $\mathcal{S}_{\mathbf{HAP}} \models_{\mathbf{EXP}} \mathcal{G}$ *then* $\exists \mathbf{EXP}'$ *such that* $\mathcal{S}_\emptyset \vdash^{\mathbf{HAP}}_{\mathbf{EXP}'} G$ *with an expectation answer* $(\mathbf{EXP}', \sigma)$ *such that* $\mathbf{EXP}'\sigma \subseteq \mathbf{EXP}$*.*

### 3.1.3 Runtime verification examples

Let us consider the following case: *telco* sends the bill, and $C$ does not pay. As a consequence, after $TWait$ time units *telco* sends $C$ a request for payment.

$$\begin{array}{l} \mathbf{H}(\ tell(\ telco,\ c,\ phone\_bill(390512093086, 145886, 205),\ 19). \\ \mathbf{H}(\ tell(\ telco,\ c,\ request\_payment(390512093086, 145886, 205),\ 33). \\ \mathbf{H}(\ tell(\ c,\ telco,\ pay(390512093086, 145886, 205, 1674521),\ 37). \end{array} \tag{10}$$

This sequence of events (10) generates a set of fulfilled expectations. What happens is, after the first message at time 19 (the notification of the *phone_bill*), $[IC2]$ generates three alternative and equally plausible sets of expectations: either $C$ is expected to pay before time 29, or $C$ is expected to complain before time 29, or else *telco* has the right ($\neg\mathbf{EN}$) to issue a request for payment after time 29. In all cases *telco* does not have the right to send a request for payment before time 29, because of $[IC1]$. At time 29 the first two alternatives become invalid due to the expired deadline. The message *request_payment* at time 33 is indeed acceptable, according to the contract, and it gives *telco* explicit right to de-activate the carrier any time later than 29. In particular, by $[IC3]$, it generates a new choice point in the tree of expectation sets: in one case *telco* has the right to de-activate the carrier after time 39, in the other case $C$ is expected to pay. Because of $[IC4]$, the last message, in which $C$ notifies his payment to *telco*, has as a side effect that *telco* loses its right to de-activate the carrier at any time in connection to the bill No. *145886*.

As the second example shows (11), a violation can be generated if *telco* de-activates the carrier. In that case, $\mathcal{S}$CIFF detects a violation because the

fourth message violates the contract, and in particular $[IC4]$, by which *telco* is expected not to de-activate the carrier if $C$ pays within 10 time units after receipt of *telco*'s request for payment.

$$\left.\begin{array}{l}\mathbf{H}(\ tell(\ telco,\ c,\ phone\_bill(390512093086, 145886, 205),\ 19).\\ \mathbf{H}(\ tell(\ telco,\ c,\ request\_payment(390512093086, 145886, 205),\ 33).\\ \mathbf{H}(\ tell(\ c,\ telco,\ pay(390512093086, 145886, 205, 1674521),\ 37).\end{array}\right] (10)$$
$$\mathbf{H}(\ tell(\ telco,\ c,\ de\_activate(390512093086, reason(145886)),\ 38). \quad (11)$$

Let us consider a third example, starting by *telco* sending $C$ a bill, as in all other examples. $C$ complains, but he does it at time 33, which unfortunately is after the deadline of 10 time units after the bill. This complaint, although not specifically disallowed by the contract, does not change the state of expectations in the system, since no IC fires. In particular, $[IC5]$ says that if $C$ complains before the deadline, he is not expected any more to pay the amount he complained about, and *telco* looses the right to send requests for payment concerning either the amount $C$ complained about or concerning the full amount of the bill. But $[IC5]$ (as well as the other ICs) does not say what happens in case of a late complaint. *telco* therefore sends him a request to payment, since it is its right, and the only options for $C$ are either to pay, or to have the carrier de-activated. $C$ pays and *telco* has no more right to de-activate the line, which incidentally makes that second option (have the carrier de-activated) inconsistent, besides fulfilling all the expectations of the first branch (12).

$$\begin{array}{l}\mathbf{H}(\ tell(\ telco,\ c,\ phone\_bill(390512093086, 145886, 205),\ 19).\\ \mathbf{H}(\ tell(\ c,\ telco,\ complain(390512093086, 145886, 150),\ 33).\\ \mathbf{H}(\ tell(\ telco,\ c,\ request\_payment(390512093086, 145886, 205),\ 34).\\ \mathbf{H}(\ tell(\ c,\ telco,\ pay(390512093086, 145886, 205, 1674521),\ 37).\end{array} \quad (12)$$

In the last example, *telco* as usual sends $C$ a bill. However, this time $C$ sends his complaint before the deadline. $C$ complains about an amount of €150 out of €205. Moreover, the complaint is judged admissible (in our example, shown with the *is_admissible_complaint* predicate). As a consequence, if *telco* sends $C$ a request for payment (13), it causes a contract violation. Due to $[IC5]$, *telco* can no longer issue a request for payment. Unfortunately, *telco* does so at time 34, and consequently $\mathcal{S}$CIFF detects the violation of $[IC5]$.

$$\begin{array}{l}\mathbf{H}(\ tell(\ telco,\ c,\ phone\_bill(390512093086, 145886, 205),\ 19).\\ \mathbf{H}(\ tell(\ c,\ telco,\ complain(390512093086, 145886, 150),\ 24).\\ \mathbf{H}(\ tell(\ telco,\ c,\ request\_payment(390512093086, 145886, 205),\ 34).\end{array} \quad (13)$$

## 3.2 Design-time property verification

In order to verify contract properties, we have developed an extension of the $\mathcal{S}$CIFF proof-procedure, called g-$\mathcal{S}$CIFF [ACG+06c]. In the following, we briefly

recall g-$\mathcal{S}$CIFF, and then show how it can be used to refute a formal property that the contract described in Sect. 2.4 does not enjoy.

### 3.2.1 The g-$\mathcal{S}$CIFF proof procedure

Besides verifying whether a history is compliant to a contract, g-$\mathcal{S}$CIFF is able to generate a compliant history, given a contract. This is achieved by $(i)$ considering **H** events as abducibles and $(ii)$ adding a new transition to $\mathcal{S}$CIFF, which, when an expectation is added to the set of expectations, generates an event that fulfills it. g-$\mathcal{S}$CIFF has been proved sound [ACG$^+$05], which means that the histories that it generates (in case of success) are guaranteed to be compliant to the interaction contracts while entailing the goal. Note that the histories generated by g-$\mathcal{S}$CIFF are in general not only a collection of ground events, like the **HAP** sets given as an input to $\mathcal{S}$CIFF. They can, in fact, contain variables, which means that they represent *classes* of event histories.

In order to use g-$\mathcal{S}$CIFF for verification, we express the property to be verified as a conjunction of literals. If we want to verify if a formula $f$ is a property of a contract $\mathcal{P}$, we express the contract in our language and $\neg f$ as a g-$\mathcal{S}$CIFF goal. Then either:

- g-$\mathcal{S}$CIFF returns success, generating a history **HAP**. Thanks to the soundness of g-$\mathcal{S}$CIFF, **HAP** entails $\neg f$ while being compliant to $\mathcal{P}$: $f$ is not a property of $\mathcal{P}$, **HAP** being a counterexample; or
- g-$\mathcal{S}$CIFF returns failure, suggesting that $f$ is a property of $\mathcal{P}$.[3]

### 3.2.2 Design-time property verification example

In this section, we show the refutation, by means of g-$\mathcal{S}$CIFF, of a simple property of the contract described in Sect. 2.4.2. For simplicity, we will not show the details related to the management of restrictions and defined predicates.

The property is the following: if a phone bill is sent, then the customer will pay for it. Using our formalism for events, the property can be written as follows:

$$\begin{aligned} &\mathbf{H}(\ tell(\ T,\ C,\ phone\_bill(N,I,A),D),T_b) \\ \rightarrow&\mathbf{H}(\ tell(\ C,\ T,\ pay(N,I,A,R),D),T_p) \end{aligned} \tag{14}$$

The negation of the property is:

$$\begin{aligned} &\mathbf{H}(\ tell(\ T,\ C,\ phone\_bill(N,I,A),D),T_b) \\ \wedge&\neg\mathbf{H}(\ tell(\ C,\ T,\ pay(N,I,A,R),D),T_p) \end{aligned} \tag{15}$$

[3] If we had a completeness result for g-$\mathcal{S}$CIFF, this would indeed be a proof and not only a suggestion.

Therefore, a history that entails Eq. (15) is a counterexample of the property that we want to verify. To try and find such a history, we write the following g-$\mathcal{S}$CIFF goal:

$$\begin{aligned}&\mathbf{E}(\ tell(\ T,\ C,\ phone\_bill(N,I,A),D),T_b)\\ \wedge&\mathbf{EN}(\ tell(\ C,\ T,\ pay(N,I,A,R),D),T_p)\end{aligned} \tag{16}$$

and run g-$\mathcal{S}$CIFF. A history that achieves the goal will necessarily include events that are expected to happen, and not include events that are expected not to happen, in the goal.

g-$\mathcal{S}$CIFF imposes the first expectation of the goal,

$$\mathbf{E}(\ tell(\ T,\ C,\ phone\_bill(N,I,A),D),T_b),$$

which generates the following event:

$$\mathbf{H}(\ tell(\ T,\ C,\ phone\_bill(N,I,A),D),T_b)$$

which in turn, due to the first IC, generates the expectation

$$\mathbf{EN}(tell(T, C, request\_payment(N, I, A), D), T2)$$

and, due to the second, one of

$$\mathbf{E}(tell(C, T, pay(N, I, A, PR), D), T2),$$

$$\mathbf{E}(tell(C, T, complain(N, I, PA), D), T3),$$

$$\neg\mathbf{EN}(tell(T, C, request\_payment(N, I, A), D), T4).$$

The **E**-consistency requirement (Def. 2.10) rules out the first alternative, because of the negative (**EN**) expectation imposed by the goal (see Eq. (16)); so the second branch is explored, and the event

$$\mathbf{H}(tell(C, T, complain(N, I, PA), D), T3)$$

is generated.

Due to the fifth IC, the following expectations are generated:

$$\neg\mathbf{E}(tell(C, T, pay(N, I, PA, PR), D), T3)$$

and

$$\mathbf{EN}(tell(T, C, request\_payment(N, I, Bill\_Amnt), D), T4)$$

and finally g-$\mathcal{S}$CIFF terminates and returns success, with the history

$$\begin{aligned}\mathbf{HAP} = \{&\mathbf{H}(tell(T, C, phone\_bill(N, I, A), D), T_b),\\ &\mathbf{H}(tell(C, T, complain(N, I, PA), D), T3)\}\end{aligned}$$

Thanks to the soundness of g-$\mathcal{S}$CIFF, **HAP** is a counterexample of the property that we wanted to prove, and it is also compliant to the contract. Thus, it shows that the contract does not enjoy the property. In particular, it shows that a customer can avoid being expected to pay by filing a complaint.

# 4 Rule Mark-Up

In [ACG+06a], we propose an architecture and a formal framework that enables web services to reason on publicly available $\mathcal{S}$CIFF-based specifications: in particular, it is possible for a web service to verify whether it can interact with another and achieve a goal. We believe that an interested party could fruitfully perform such a step before agreeing with another party on a contract. Obviously, this requires a formalism that makes it practical to exchange $\mathcal{S}$CIFF-based specifications.

RuleML [AST05] is the perfect mark-up language for exchanging rules on the web, so our choice has been easy. RuleML 0.9 contains mark-ups for expressing important concepts of the $\mathcal{S}$CIFF proof-procedure. In particular, $\mathcal{S}$CIFF is a rule engine able to distinguish and use both backward and forward rules. Backward rules are used to plan, reason upon events, perform proactive reasoning. Forward rules are used for reactive reasoning, to quickly perform actions in response to occurred events. Both are seamlessly integrated in $\mathcal{S}$CIFF. RuleML 0.9 contains a *direction* attribute that can be attached to rules. Being based on abduction, $\mathcal{S}$CIFF can deal both with negation as failure and negation by default, that have an appropriate tagging in RuleML. In this work, we only used standard RuleML syntax; in future work we might be interested in distinguishing between defined and abducible predicates, or between expectations and events.

$\mathcal{S}$CIFF was implemented in SICStus Prolog: SICStus contains an implementation of the PiLLoW library [GH01], which makes it easy to perform http requests, as well as implementing services on the web. Finally, SICStus contains an XML parser, which allowed us to easily implement the RuleML parser. The RuleML parser is freely available on the $\mathcal{S}$CIFF web site [SCI05].

# 5 Related work

The reduction of deontic concepts such as obligations and prohibitions has been the subject of several past works: notably, by [And58] (according to which, informally, $A$ is obligatory iff its absence produces a state of violation) and by [Mey88] (where, informally, an action $A$ is prohibited iff its being performed produces a state of violation). These two reductions strongly resemble our definition of fulfillment (Def. 2.11), which requires positive (resp. negative) expectations to have (resp. not to have) a corresponding event.

Several papers discuss "sub-ideal" situations, i.e., how to manage situations in which some of the norms are not respected.

For instance, [vT99] shows the relation between diagnostic reasoning and deontic logic, importing the *principle of parsimony* from diagnostic reasoning into their deontic system, in the form of a requirement to minimize the number of violations. In particular, given the specification of a normative system (as a set of formulae which tell when a norm is violated) and a state of affairs, they define a minimal (with respect to inclusion) set of norms such that the violation

of those norms is consistent with the specification and the state of affairs. The SOCS social framework, currently, only distinguishes between empty and non-empty sets of violations, and does not define minimal sets. However, it would be possible to do so by taking the minimal, with respect to inclusions, among the sets of expectations which are consistent with a social specification and a history, but possibly not fulfilled by the history. This will probably be our approach when we tackle the management of violations (by means of sanctions and recovery procedures) in future work.

[PS96] propose a solution to the problem and paradoxes stemming from earlier logical representations of *contrary-to-duty* obligations, i.e., obligations that become active when other obligations are violated. They do so by introducing a new operator $O_B(A)$, meaning that $A$ is obligatory given the sub-ideal context $B$. The semantics of this operator is of Kripke type, but it differs to the standard modal logic because of the accessibility relation: in that work, the accessible worlds are the best alternatives, given the truth of $B$. In the "main stream" of our research, we do not support contrary-to-duty obligations. However, we proposed a modified version of our framework [ADG$^+$04] , which provides a simplified language and does support alternative obligations at different levels of priority; a further step could be to integrate priority levels in the main SOCS social framework.

Deontic operators have been used not only at the social level, but also at the agent level. Notably, in IMPACT ([AOR$^+$99, ESP99]), agent programs may be used to specify what an agent is obliged to do, what an agent may do, and what an agent cannot do on the basis of deontic operators of Permission, Obligation and Prohibition (whose semantics does not rely on a Deontic Logic semantics). In this respect, the IMPACT and SOCS social models have similarities even if their purpose and expressivity are different. The main difference is that the goal of agent programs in IMPACT is to express and determine by its application the behavior of a single agent, whereas the SOCS social model goal is to express rules of interaction and norms, that instead cannot really determine and constrain the behavior of the single agents participating to a society, since agents are autonomous.

Governatori [Gov05] uses defeasible logics with deontic operators of *O*bligation and *P*ermission to define contracts. He proposes the introduction in RuleML of new tags for identifying obligations and permission. In $\mathcal{S}$CIFF, we usually do not use explicit permission, because everything is allowed by default; we typically state explicitly when an action is expected not to happen **EN**. There are connections between **EN** and $\neg P$ of deontic logics (studied in [AGL$^+$06]), so we might use the same tags proposed by Governatori (e.g., we might use `<neg><Permission>` to represent **EN**).

Governatori [Gov05] also introduces an operator $\otimes$ to address recovery from violation. For example, $A \Rightarrow OB \otimes OC$ means that $A$ implies that $B$ is obligatory; however in case $OB$ is violated, $C$ becomes obligatory. In $\mathcal{S}$CIFF, recovery expectations can be inserted as an alternative in each of the rules: $A \Rightarrow OB \otimes OC$ could be written in $\mathcal{S}$CIFF as $\mathbf{H}(A) \rightarrow \mathbf{E}(B) \vee \mathbf{E}(C)$. Interestingly, Governatori [Gov05] proposes also an inference rule that derives recovery rules from the other

rules of the contract (from $A \rightarrow OB$ and $\neg B \rightarrow OC$ derives $A \rightarrow OB \otimes OC$); this is an interesting line of research that we plan to apply in future work also to $\mathcal{S}$CIFF.

In cite [GM05, GM06] Governatori and Milosevic discuss the need for contract verification and contract monitoring to check how parties fulfil their policies. Both these issues are addressed by the adoption of a formal specification language for contracts. The system they propose, and their Business Contract Language (BCL) in particular, is based on the formalism for the representation of contrary-to-duty obligations (CTDs), i.e., obligations in force after some other obligations have been violated. The formal representation for contracts they adopt is based upon a propositional logic language, with the deontic operators of obligation, permission and contrary-to-duty. Each condition or policy of a contract is represented by a rule where the *antecedent* is a literal or a modal literal (built with the deontic operators of permission and obligation, possibly negated) and the *conclusion* of the rule is a CTD expression. Contract analysis is then done by reducing a contract to a normal form, where all the contract conditions that can be generated/derived from the given specification have been made explicit. The procedure to generate normal forms is expressed in terms of inference rules, which merge two rules in a new clause through the violations of conditions (e.g., when the former rule mentions an obligation $O\ A$ in its conclusion and the latter rule has the negation $\neg A$ in its antecedent, then their conclusions are composed in order to build a CDT formula for $A$). Normal forms are then a sort of *partial evaluation* of specification rules, in the logic of violation, aiming at producing rules with CDT formulas in their conclusions which summarize all the possible violations and recovery actions implicitly specified by the original (logic) representation of a contract. On generated normal forms they can therefore detect conflicts arising from, e.g., obligation of $A$ and $\neg A$, or occurrence of $A$ and $\neg A$ in conclusions without any CTD for $A$ neither $\neg A$.

Although we do not have a language supporting CDTs, our language is first-order, and supports the deontic operators of permission and obligation (and their negation as discussed in [AGL$^+$06]). In our approach, $\mathcal{S}$CIFF is exploited at run-time for contract monitoring (e.g., conflicts and contradictions are detected at run-time by the notions of $E$-consistency and $\neg$-consistency), and more general contract properties (beside the absence of conflicts) can be also statically verified by g-$\mathcal{S}$CIFF. g-$\mathcal{S}$CIFF, in particular, generates all the possible *compliant* histories which satisfy a given goal, and a contract specified in the $\mathcal{S}$CIFF language. Each generated history can be considered as a set of obligations in the approach of Governatori and Milosevic [GM06], since g-$\mathcal{S}$CIFF turns obligations into events.

An interesting extension would be to equip the $\mathcal{S}$CIFF language with CTDs expressions, to occur in the head of ICs. This is subject for future work.

[Bv03] discuss how a normative system can be seen as a normative agent, equipped with mental attitudes, about which other agents can reason, choosing either to fulfill their obligations, or to face the possible sanctions. Conceptually, the social infrastructure in the SOCS model could be viewed as an agent, whose knowledge base is the society specification, whose mental attitude is a set of

expectations, and whose reasoning process is the $\mathcal{S}$CIFF proof procedure.

[BDDM04] investigate the deontic logic of deadlines by introducing an operator $O(\rho \leq \delta)$, which means, intuitively, that the action $\rho$ ought to be brought about before (or at the same time) another event $\delta$ happens. They model time by means of the CTL temporal logic. We can express a similar concept by means of an integrity constraints $\mathbf{H}(\delta, T_\delta) \rightarrow \mathbf{E}(\rho, T_\rho) \wedge T_\rho \leq T_\delta$, which says that, if $\delta$ has happened, than $\rho$ is expected to have happened before (or at the same time).

The $\mathcal{S}$CIFF framework can capture, in a computational setting, the concept of (conditional) obligation with deadline presented by [DMDW02], with an explicit mapping of time. Dignum *et al.* write: `Oa(r<d|p)` to state that if the precondition `p` becomes valid, the obligation becomes active. The obligation expresses the fact that `a` is expected to bring about the truth of `r` before a certain condition `d` holds.

For instance, if we have:

$$
\begin{aligned}
p &= \mathbf{H}(tell(S, a, request(G), D, T)) \\
r &= \mathbf{H}(tell(a, S, answer(G), D, T')), T' > T \\
d &= T' > T + 2
\end{aligned}
$$

we can map `Oa(r<d|p)` into a IC:

$\mathbf{H}(tell(S, a, request(G), D), T) \rightarrow$
$\mathbf{E}(tell(a, S, answer(G), D), T'), T' > T, T' \leq T + 2.$

There have been many works using the Event Calculus (EC) for the purpose of reasoning over the effects of events, that are very close to this paper. In particular, our work is very related to the work in [FSSB05]. In [FSSB05] the authors have been principally concerned with the representation of contracts and their normative state in particular, in terms of obligation, power and permission. The effects of contract events on the normative state of a contract are specified using an XML formalisation of the Event Calculus. This representation may be used to track the state of the agreement, according to a narrative of contract events similar to our concept of History.

Like in [FSSB05], $\mathcal{S}$CIFF can be seen as a generic language for expressing backward and forward rules and reasoning about (conformance) properties of a specific where the representation of contracts is just one application.

Differently from [FSSB05] In this work we show that being able to describe contracts as logical theories is extremely useful not only for tracking, but also for for proving general or specific properties of the contracts by using the same formalism . A similar approach is used in [ASP03] by using a formalization in terms of transition systems and model checking techniques.

# 6 Conclusions

In this paper, we proposed the use of the $\mathcal{S}$CIFF framework, originally developed for agent interaction protocols, to specify and verify business contracts. We supported intuitively our proposal by showing a deontic reading of $\mathcal{S}$CIFF specifications. We gave the specification of sample business contract clauses in the $\mathcal{S}$CIFF language.

We also demonstrated how verification is performed in the $\mathcal{S}$CIFF framework: in particular, run-time verification by means of the $\mathcal{S}$CIFF proof procedure, and design-time property verification with the g-$\mathcal{S}$CIFF proof procedure. We also showed how $\mathcal{S}$CIFF rules can be encoded in RuleML, in order to be possibly exchanged to enable potential contract parties to reason on contracts in advance.

Future work will be devoted to experiment with the $\mathcal{S}$CIFF framework on real-world contracts, to test both the expressiveness of the $\mathcal{S}$CIFF language and the effectiveness of the proof procedures used for verification. We are also working on a formal completeness result (possibly for restricted cases) for g-$\mathcal{S}$CIFF.

# Acknowledgments

This work has been partially supported by the MIUR PRIN 2005 projects *Specification and verification of agent interaction protocols* and *Vincoli e preferenze come formalismo unificante per l'analisi di sistemi informatici e la soluzione di problemi reali*, and by the MIUR FIRB project *Tecnologie Orientate alla Conoscenza per Aggregazioni di Imprese in Internet*.

# References

[AB91] Krzysztof R. Apt and Marc Bezem. Acyclic programs. *New Generation Computing*, 9(3/4):335–364, 1991.

[AB94] Krzysztof R. Apt and Roland N. Bol. Logic programming and negation: A survey. *Journal of Logic Programming*, 19/20:9–71, 1994.

[ACG+05] Marco Alberti, Federico Chesani, Marco Gavanelli, Evelina Lamma, Paola Mello, and Paolo Torroni. On the automatic verification of interaction protocols using *g-*$\mathcal{S}$CIFF. Technical Report DEIS-LIA-04-004, University of Bologna (Italy), 2005. LIA Series no. 72.

[ACG+06a] Marco Alberti, Federico Chesani, Marco Gavanelli, Evelina Lamma, Paola Mello, Marco Montali, and Paolo Torroni. Policy-based reasoning for smart web service interaction. In *Proceedings of the 1st International Workshop on Applications of Logic Programming in the Semantic Web and Semantic Web Services (ALPSWS*

*2006)*, volume 196 of *CEUR Workshop Proceedings*, pages 87–102, Seattle, WA, USA, August 2006.

[ACG+06b] Marco Alberti, Federico Chesani, Marco Gavanelli, Evelina Lamma, Paola Mello, and Paolo Torroni. Compliance verification of agent interaction: a logic-based tool. *Applied Artificial Intelligence*, 20(2-4):133–157, February-April 2006.

[ACG+06c] Marco Alberti, Federico Chesani, Marco Gavanelli, Evelina Lamma, Paola Mello, and Paolo Torroni. Security protocols verification in abductive logic programming: a case study. In Ogus Dikenelli, Marie-Pierre Gleizes, and Alessandro Ricci, editors, *ESAW 2005 Post-proceedings*, number 3963 in LNAI, pages 106–124, Kusadasi, Aydin, Turkey, 2006. Springer-Verlag.

[ACG+06d] Marco Alberti, Federico Chesani, Marco Gavanelli, Evelina Lamma, Paola Mello, and Paolo Torroni. Verifiable agent interaction in abductive logic programming: the $\mathcal{S}$CIFF proof-procedure. Technical Report DEIS-LIA-06-001, University of Bologna (Italy), March 2006. LIA Series no. 75.

[ADG+04] Marco Alberti, D. Daolio, Marco Gavanelli, Evelina Lamma, Paola Mello, and Paolo Torroni. Specification and verification of agent interaction protocols in a logic-based system. In Hisham M. Haddad, Andrea Omicini, and Roger L. Wainwright, editors, *Proceedings of the 19th Annual ACM Symposium on Applied Computing (SAC 2004). Special Track on Agents, Interactions, Mobility, and Systems (AIMS)*, pages 72–78, Nicosia, Cyprus, March 14–17 2004. ACM Press.

[AGL+05] Marco Alberti, Marco Gavanelli, Evelina Lamma, Paola Mello, and Paolo Torroni. The $\mathcal{S}$CIFF abductive proof-procedure. In *Proceedings of the 9th National Congress on Artificial Intelligence, AI*IA 2005*, volume 3673 of *Lecture Notes in Artificial Intelligence*, pages 135–147. Springer-Verlag, 2005.

[AGL+06] Marco Alberti, Marco Gavanelli, Evelina Lamma, Paola Mello, Giovanni Sartor, and Paolo Torroni. Mapping deontic operators to abductive expectations. *Computational and Mathematical Organization Theory*, 12(2–3):205 – 225, October 2006.

[AMB00] Grigoris Antoniou, Michael J. Maher, and David Billington. Defeasible logic versus logic programming without negation as failure. *J. Log. Program.*, 42(1):47–57, 2000.

[And58] A. Anderson. A reduction of deontic logic to alethic modal logic. *Mind*, 67:100–103, 1958.

[AOR+99] K. A. Arisha, F. Ozcan, R. Ross, V. S. Subrahmanian, T. Eiter, and S. Kraus. IMPACT: a Platform for Collaborating Agents. *IEEE Intelligent Systems*, 14(2):64–72, March/April 1999.

[ASP03] Alexander Artikis, Marek J. Sergot, and Jeremy Pitt. An executable specification of an argumentation protocol. In *ICAIL*, pages 1–11, 2003.

[AST05] Asaf Adi, Suzette Stoutenburg, and Said Tabet, editors. *Rules and Rule Markup Languages for the Semantic Web, First International Conference, RuleML 2005, Galway, Ireland, November 10-12, 2005, Proceedings*, volume 3791 of *Lecture Notes in Computer Science*. Springer Verlag, 2005.

[BDDM04] Jan Broersen, Frank Dignum, Virginia Dignum, and John-Jules Ch. Meyer. Designing a deontic logic of deadlines. In Alessio Lomuscio and Donald Nute, editors, *DEON*, volume 3065 of *Lecture Notes in Computer Science*, pages 43–56. Springer, 2004.

[Bür94] H.J. Bürckert. A resolution principle for constrained logics. *Artificial Intelligence*, 66:235–271, 1994.

[Bv03] Guido Boella and Leendert W. N. van der Torre. Attributing mental attitudes to normative systems. In J. S. Rosenschein, T. Sandholm, M. Wooldridge, and M. Yokoo, editors, *Proceedings of the Second International Joint Conference on Autonomous Agents and Multiagent Systems (AAMAS-2003)*, pages 942–943, Melbourne, Victoria, July 14–18 2003. ACM Press.

[Cla78] K. L. Clark. Negation as Failure. In H. Gallaire and J. Minker, editors, *Logic and Data Bases*, pages 293–322. Plenum Press, 1978.

[DMDW02] V. Dignum, J. J. Meyer, F. Dignum, and H. Weigand. Formal specification of interaction in agent societies. In *Proceedings of the Second Goddard Workshop on Formal Approaches to Agent-Based Systems (FAABS), Maryland*, October 2002.

[DS98] M. Denecker and D. De Schreye. SLDNFA: an abductive procedure for abductive logic programs. *Journal of Logic Programming*, 34(2):111–167, 1998.

[ESP99] T. Eiter, V.S. Subrahmanian, and G. Pick. Heterogeneous active agents, I: Semantics. *Artificial Intelligence*, 108(1-2):179–255, March 1999.

[FK97] T. H. Fung and R. A. Kowalski. The IFF proof procedure for abductive logic programming. *Journal of Logic Programming*, 33(2):151–165, November 1997.

[FSSB05] Andrew D. H. Farrell, Marek J. Sergot, Mathias Sallé, and Claudio Bartolini. Using the event calculus for tracking the normative state of contracts. *Int. J. Cooperative Inf. Syst.*, 14(2-3):99–129, 2005.

[GH01] Daniel Cabeza Gras and Manuel V. Hermenegildo. Distributed WWW programming using (Ciao-)Prolog and the PiLLoW library. *Theory and Practice of Logic Progr.*, 1(3):251–282, 2001.

[GLC99] Benjamin N. Grosof, Yannis Labrou, and Hoi Y. Chan. A declarative approach to business rules in contracts: courteous logic programs in xml. In *ACM Conference on Electronic Commerce*, pages 68–77, 1999.

[GM05] Guido Governatori and Zoran Milosevic. Dealing with contract violations: formalism and domain specific language. In *EDOC*, pages 46–57. IEEE Computer Society, 2005.

[GM06] Guido Governatori and Zoran Milosevic. A formal analysis of a business contract language. *International Journal of Cooperative Information Systems*, 15(4):659–685, 2006.

[Gov05] Guido Governatori. Representing business contracts in RuleML. *International Journal of Cooperative Information Systems*, 14(2-3):181–216, 2005.

[JM94] J. Jaffar and M.J. Maher. Constraint logic programming: a survey. *Journal of Logic Programming*, 19-20:503–582, 1994.

[Kun87] K. Kunen. Negation in logic programming. In *Journal of Logic Programming*, volume 4, pages 289–308, 1987.

[Mey88] J. J. Ch. Meyer. A different approach to deontic logic: Deontic logic viewed as a variant of dynamic logic. *Notre Dame J. of Formal Logic*, 29(1):109–136, 1988.

[MGL+04] Z. Milosevic, S. Gibson, P. F. Linington, J. Cole, and S. Kulkarni. On design and implementation of a contract monitoring facility. In *Proceedings of the First International Workshop on Electronic Commerce (WEC'04)*, pages 62–70, Los Alamitos, CA, USA, 2004. IEEE Computer Society.

[Mil95] Zoran Milosevice. *Enterprise aspects of open distributed systems.* PhD thesis, Computer Science Department, The University of Queensland, October 1995.

[PS96] Henry Prakken and Marek Sergot. Contrary-to-duty obligations. *Studia Logica*, 57(1):91–115, 1996.

[Sar04] Giovanni Sartor. *Legal Reasoning*, volume 5 of *Treatise*. Kluwer, Dordrecht, 2004.

[SCI05] The $\mathcal{SCIFF}$ abductive proof procedure, 2005. http://lia.deis.unibo.it/research/sciff/.

[SIC06] SICStus prolog user manual, release 3.12.7, October 2006. http://www.sics.se/isl/sicstus/.

[SOC05] Societies Of ComputeeS (SOCS): a computational logic model for the description, analysis and verification of global and open societies of heterogeneous computees. IST-2001-32530, 2002-2005. Home Page: http://lia.deis.unibo.it/research/socs/.

[vT99] Leendert W. N. van der Torre and Yao-Hua Tan. Diagnosis and decision making in normative reasoning. *Artificial Intelligence and Law*, 7(1):51–67, 1999.

[Xan03] I. Xanthakos. *Semantic Integration of Information by Abduction.* PhD thesis, Imperial College London, 2003. Available at http://www.doc.ic.ac.uk/~ix98/PhD.zip.

# Implementing Norms that Govern Non-Dialogical Actions

Viviane Torres da Silva*

Departamento de Sistemas Informáticos y Programación – UCM, Spain, Madrid
viviane@fdi.ucm.es

**Abstract.** The governance of open multi-agent systems is particular important since those systems are composed by heterogeneous, autonomous and independently designed agents. Such governance is usually provided by the establishment of norms that regulate the actions of agents. Although there are several approaches that formally describe norms, there are still few of them that propose their implementation. In additions, only one that provides support for implementing norms deals with non-dialogical actions since the others only deal with dialogical actions, i.e., actions that provide the interchange of messages between agents. In this paper we propose the implementation of norms that govern non-dialogical actions by extending one of the approaches that regulate dialogical ones. Non-dialogical actions are not related to the interactions between agents but to tasks executed by agents that characterize, for instance, the access to resources, their commitment to play roles or their movement into environments and organizations.

**Keywords:** norm, governance of multi-agent system, non-dialogical action, implementation of norm

## 1 Introduction

The governance of open multi-agent systems copes with the heterogeneity, autonomy and diversity of interests among agents that can work towards similar or different ends [8] by establishing norms. The set of system norms defines actions that agents are prohibited, permitted or obligated to do [1,11].

Several works have been proposed in order to define the theoretical aspects of norms [3,5], to formally define those norms [2,4], and to implement them [6,7,8,9,12]. In this paper we focus on the implementation of norms. Our goal is to present an approach where dialogical and non-dialogical norms can be described and regulated. Non-dialogical actions are not related to the interactions between agents but to tasks executed by agents that characterize, for instance, the access to resources, their commitment to play roles or their movement in environments and organizations. From the set of analyzed proposals for implementing norms, the only approach that considers non-dialogical actions is [12]. Although, it presents some issues on the verification and enforcement of norms, it does no demonstrate how they should be

* Research supported by Comunidad de Madrid S-0505/TIC-407 and MEC-SP TIC2003-01000.

Dagstuhl Seminar Proceedings 07122
Normative Multi-agent Systems
http://drops.dagstuhl.de/opus/volltexte/2007/927

implemented. Other approaches such as [6,7,8,9] deal with e-Institutions and, thus, consider illocutions as the only action performed in such systems.

Our approach extends the work presented in [7] with the notion of non-dialogical actions proposed in [12]. A normative language is presented in [7] to describe illocutions (dialogical actions) that might be dependent on temporal constraints or the occurrence of events. We have extended the normative language in order to be possible to specify non-dialogical norms that state obligations, permissions or prohibition over the execution of actions of agents' plans (as proposed in [12]) and of object methods. Similar to the approach presented in [7], we have also used Jess[1] to implement the governance mechanism that regulates the behavior of agents. The mechanism activates norms and fires violations (Jess rules) according to the executed (dialogical or non-dialogical) actions (Jess facts).

The paper is organized as follows. Section 2 describes the example we are using to illustrate our approach. Section 3 intends to clearly present the difference between dialogical and non-dialogical actions. Section 4 points out the main concepts of the extended normative language and Section 5 describes the implementation of the governance engine in Jess. Section 6 concludes our work.

## 2 Applied Example

In order to exemplify our approach, we have defined a set of six norms that govern a simplified version of a soccer game. The soccer game is composed of agents playing one of the three available roles: referee, coach and player (kicker or goalkeeper). The responsibilities of a referee in a soccer game are: to start the game, stop it, check the players' equipments and punish the players. The available punishments are: to show a yellow card, send off a player, and declare a penalty. The possible actions of a player during a game are: kick the ball and get the ball with hands. The coach role is limited to substitute players. Besides those actions, all agents are able to move and, therefore, enter and leave the game filed. The six norms that regulate our simple soccer game are the following:

***Norm 1:*** *The referee must check the players' equipments before star the game.*
***Norm 2:*** *A coach cannot substitute more than three players in the same game.*
***Norm 3:*** *Players cannot leave the game field during the game.*
***Norm 4:*** *The referee must send off a player after (s)he has done a second caution in the same match.* In this simplified version of the soccer game, there is only one situation that characterizes a caution; a player leaving the game field before the referee has stopped it. At the first caution, the agent receives a yellow card.
***Norm 5:*** *Kickers cannot get the ball.*
***Norm 6:*** *The referee must declare a penalty if kicker gets the ball.*

[1] Jess is a rule-based system. http://www.jessrules.com/

## 3 Dialogical x Non-Dialogical Actions

Non-dialogical actions are the ones not related to interactions between agents. Not all actions executed by agents in multi-agent systems provide support for sending and receiving messages between them [12]. There are actions that modify the environment (for example, updating the state of a resource) that do not characterize a message being sent to or received from another agent. In the soccer game example, the actions of kicking the ball or getting it are non-dialogical actions. In addition, actions that modify the position of an agent in an environment do not characterize a dialogical action either. The actions of entering or leaving the game field are not dialogical ones.

Some actions can be defined as a dialogical or a non-dialogical one, depending on how the problem is modeled. In the soccer game, to start a game and to stop it was considered dialogical actions. Agents receive a message informing about the state of the game. The dialogical actions of the soccer game example are: to start the game, stop it, punish player, declare penalty and show the yellow card. The non-dialogical ones are: enter in the game filed, leave it, get the ball, kick the ball, substitute a player and check the player's equipment.

## 4 Describing Norms

Since our intention is to contribute to the work presented in [6], we extend the BNF normative language to represent non-dialogical actions and to describe conditions and time situations that are defined by those non-dialogical actions. In addition, the specification of dialogical actions already presented in the previous normative language was extended in order to be possible to describe messages attributes stated in the FIPA ACL language[2].

### 4.1 Specifying Non-Dialogical Actions

The original BNF description of the normative language defines norms as the composition of a *deontic* concept (characterizing obligation, prohibition or permission) and an action followed by a temporal situation and a *if* condition, when pertinent. In such definition, actions are limited to utterance of illocutions.

In our proposed extension, the *action* concept was generalized to also describe non-dialogical ones. Non-dialogical actions define the entities whose behavior is being restricted and the actions that are being regulated. Due to the way the *entity* concept was defined, a non-dialogical norm can be applied to all agents in the system, to a group of agents, to agents playing a given role or even to a unique agent.

```
<norm> ::= <deontic_concept> '(' <action> ')'
 | <deontic_concept> '(' <action><temporal_situation> ')'
 | <deontic_concept> '(' <action> IF <if_condition> ')'
 | <deontic_concept> '(' <action> <temporal_situation> IF <if_condition> ')'
<deontic_concept> ::= OBLIGED | FORBIDDEN | PEMITTED
```

[2] http://www.fipa.org/repository/aclspecs.html

```
<action>::= <non_dialogical_action> | <dialogical_action>
<non_dialogical_action> ::= <entity> 'EXECUTE' <exec>
<entity>::= <agent>':'<role> | <role> | <agent> | <group> | 'ALL'
```

In this paper we are limiting non-dialogical actions to the execution of an object/class method or to the execution of the action of an agent plan [12]. Non-dialogical norms that regulate the access to resources specify the entities that have restricted access to execute the methods of the resource. Non-dialogical norms that regulate (non-dialogical) actions not related to the access to resources describe entities that have restricted access to the execution of an action of a plan.

```
<exec> ::= <objectORclass>'.'<method>'('<parameters>')''('<contract>')'
 | <plan>':'<action>'('<parameters>')''('<contract>')'
 |...!the parameters and the contract can be omitted
```

In [12], the authors affirm that non-dialogical actions can be described as abstract actions that are not in the set of actions defined by the agents or in the set of methods of the classes. Agents must translate the actions and methods to be executed into more abstract ones. With the aim to help agents in such transformation, we propose the use of contracts. A contract is used to formally describe the behavior of the actions/methods while specifying its invariants, pre and post-conditions [10]. We do not impose any language to be used to describe the terms of a contracts.

```
<contract> ::= <pre>';'<post>';'<inv>
 |... !pre, post and inv concepts can be omitted
<pre> ::= <expression> | <expression> <opl> <pre> | <expression> ',' <pre>
... !pre, post and inv are similarly defined
<opl> ::= 'AND' | 'OR' | 'XOR' | 'NOR' |...
```

Such extensions make possible to describe, for instance, norms that regulates the execution of an action while describing the parameters required for its execution and the contract that defines it. The extensions enable, for example, the definition of *norm 2*. The norm states that a coach cannot substitute more than three players in the same game. The coach cannot execute an action that substitutes players if the number of substitutions is already 3.

```
FORBIDDEN ( coach EXECUTE managingTeam:SubstitutePlayer (outPlayer,inPlayer,team)
            ( team.coach = coach; team.substitutions = team.substitutions@pre+1,
              team.playersInField->excludes(outPlayer),
              team.playersInField->includes(inPlayer); )
 IF team.substitutions >= 3 )
```

The action governed by *norm 1* is also a non-dialogical action and states that the referee must check the players' equipment before start the game. The action of checking the equipment is a non-dialogical action since the referee needs not to interact with the player but with its equipment. On the other hand, the action of starting a game is a dialogical action modeled as a message from the referee to everybody in the game (as presented in Section 4.4).

```
OBLIGED ( referee EXECUTE managingGame:checkEquipment (players)
 BEFORE ( UTTER(game; s_i; INFORM(;referee;;[;gameStart;;;;;;])))  )
```

### 4.2 Extending the Temporal Situations

The *temporal situation* concept specified in the normative language is used to describe the period of valid (or active) norms. Norms can be activated or deactivated due to the execution of an (dialogical or non-dialogical) action, to the change in the state of an object or an agent, to the occurrence of a deadline, and to the combination of such possibilities. In the preview normative languages the authors only consider the execution of dialogical actions and the occurrence of a deadline as temporal situations. The normative language was extended to contemplate the activation and deactivation of norms due to the execution of non-dialogical actions, to the change in the state of an object or an agent (without specifying the action that was responsible for that) and to the combination of the above mentioned factors (as specified in the *situation* concept).

```
<temporal_situation> ::= BEFORE <situation> | AFTER <situation>
| BETWEEN '(' <situation> ',' <situation> ')'
```

The extensions enable, for example, the definition of *norm 3* that states that players cannot leave the game between its initial and its interruption, as shown below.

```
FORBIDDEN ( player EXECUTE moving:LeaveField ()
                          ( agent.position@pre=inField; agent.position<>inField; )
 BETWEEN ( UTTER(game; si; INFORM(;referee;;[;gameStart;;;;;;])),
           UTTER(game; si; INFORM(;referee;;[;gameStopped;;;;;;])) ))
```

Another norm that makes use of temporal situation is *norm 4*. It states that the referee must send off a player after he receives a second caution in the same match. If player leaves the field of play and it has already been shown a yellow card, the referee must send him(her) off. Note that such *norm 4* is conditioned to the execution of an action governed by *norm 3*.

```
OBLIGED ( UTTER(game;si;CAUTION(;referee;;kicker[;sentOff;;;;;;movingLeaveField]))
  AFTER ( player EXECUTE moving:LeaveField()
                          ( agent.position@pre=inField;agent.position<>inField; )
          BETWEEN ( UTTER(game; si; INFORM(;referee;;[;gameStart;;;;;;])),
                    UTTER(game; si; INFORM(;referee;;[;gameStopped;;;;;;])) ))
  IF player.yellowCard = true )
```

### 4.3 Extending the IF Condition

The *if condition* defined in the original normative language is used to introduce conditions over variables, agents' observable attributes or executed dialogical actions. Therefore, by using such language it is not possible to describe *nom 6* since it is conditioned to the execution of a non-dialogical action. Our proposed extension makes possible to specify a condition related to an executed non-dialogical action or to a fired norm.

```
<if_condition> ::= <cond_expression> | NOT '(' <cond_expression> ')'
<cond_expression> ::= <condition> | NOT <condition>
 | <condition> ',' <if_condition> | NOT <condition> ',' <if_condition>
<condition> ::= <action> | <deontic_concept> '(' <action> ')' |...
```

*Norm 6* defines that the referee must declare a penalty if a kicker gets the ball. The non-dialogical action of getting the ball is the *if condition* of *norm 6* and can be described as follows.

```
OBLIGED (UTTER(game; si; PENALTY(;referee;kickerTeam;
[;penalty;;;;;;ballTouch])) IF kicker EXECUTE play:getBall)
```

### 4.4 Extending Dialogical Actions

In [7], the authors represent the execution of dialogical actions by the identification of the action (not carried out yet) of submitting an illocution. In their point of view, an illocution is an information that caries a message to be sent by an agent playing a role to another agent playing another role. The *illocution* concept was extended to be possible to omit the agents that send and receive the messages. Not always will be possible to specify the agents that will send and receive the messages while describing the norms. Sometimes only the roles that those agents will be playing can be identified. Moreover, the roles of the sender and receiver can also be omitted. It may be the case that no mater the one is sending a message or no mater the one is receiving it, the norm must be obeyed.

```
<dialogical_action> ::= 'UTTER(' <scene> ';' <state> ';' <illocution> ')'
| 'UTTERED(' <scene> ';' <state> ';' <illocution> ')'
<illocution> ::= <perf>'('<sender>';'<role>';'<receiver> ';'<role>'['<msg>'])'
|...!it is possible to omit the senders, receivers and also their roles
```

Since a message can be sent to several agents, the *receiver* concept was also extended to make possible to describe the group of agents that will be the receivers of the message.

```
<sender> ::= <agent>
<receiver> ::= <agent> | <group>
```

By using the extensions provided above for illocution, it is possible to model *norms 1* (Section 4.1), *4* (Section 4.2) and *6* (Section 4.3) that omit the agent identification that is playing the referee role. In such cases, it is not important to identify the agent but only the role that the agent is playing. *Norm 1* also omits the receiver and its role to characterize that the message is being broadcasted. *Norm 4* identifies the role of the receiver but does not identify the agent playing the role since the message to be send does not depend on the agent. Moreover, *norm 6* does not identify the receiver agent but the receiver *team* that will be punished.

### 4.5 Specifying Messages

The *message* concept has not been specified in the previous version of the normative language. We propose to specify such concept since it may be necessary to provide some characteristics of the messages while describing the norms. The *message* concept was extended according to the parameters defined by an ACL message.

```
<msg> ::= <conversation_id>';'<contents>';'<language_encoding>';
'<ontology_protocol>';'<reply_by>';'<reply_to>';'<reply_with>';'<in_reply_to>
```

```
|...!it is possible to omit any parameter.
```

While describing *norms 4* and *6* we have used the extended *message* concept. When a referee penalizes a player it is important to inform such player why he/she is receiving such punishment. In order to provide such information we have used the `<in_reply_to>` parameter.

## 5 Implementing Norms

Once we have seen how norms can be described, we need to demonstrate how they are implemented. Similar to the approach presented in [7], we have also used Jess to implement the governance mechanism. Jess is a rule-based system that maintains a collection of facts in its knowledge base. Jess was chosen due three main reasons: (i) it provides interfaces to programs in Java (the multi-agent systems) that can use the knowledge base and declarative rules; (ii) it is possible to dynamically change the set of rules defined in Jess during the execution of Java programs and(iii) it facilitates the extensions we are proposing since the original implementation was also done in Jess.

The use of Jess makes possible to describe facts and rules that are fired according to the stated facts. In our approach, facts are agents' observable attributes, (dialogical and non-dialogical) actions executed by the agents, the norms activated by the rules, and the information about norm violations. The rules are fired according to the executed actions or observable attributes and can activate norms or assert violations.

### 5.1 The Use of Jess

In Jess, facts are described based on templates that specify the structure of the facts. We have defined a template to define agents' observable attributes and three templates to describe actions: one for describe dialogical actions and two for describing the two different kinds of non-dialogical actions contemplated in the paper (method calling and execution of the action of an agent plan). Besides, we have also described nine templates for describing each of the three norm kinds (obliged, permitted and forbidden) associated with the three different actions (message, method calling and plan execution). In addition, one template was defined for being used to describe norm violations. Such template points out the norm that was violated and the facts that have violated the norm. The two examples below illustrate templates to describe an obligation norm to execute the action of a plan and a violation.

```
(deftemplate OBLIGED-non-dialogical-action-plan
    (slot entity)(slot role)(slot plan) (slot action) (slot attribs (type String))
    (slot contract-pre (type String)) (slot contract-post (type String))
    (slot contract-inv (type String)) (slot beliefUpdated (type String))
    (slot condition (type String)))

(deftemplate VIOLATION (slot norm-violated) (multislot action-done))
```

Rules are composed by two parts. The left-hand side of the rule describes patterns of facts that need to be inserted in the knowledge base in order to fire the rule. The right-hand side defines facts that will be upload to the knowledge based if the rule is

fired. In our approach, these facts will be norms or norms' violations. Examples of rules are presented in Sections 5.3, 5.4, 5.5 and 5.6.

### 5.2 Some Guidelines

For each application norm, there is (usually) a need for describing three rules in Jess. The first rule is used to state the norm by conditioning it to the facts that activate the norm. If the facts are inserted into the knowledge based, the rule is fired and the norm is activated. The second rule deactivates the norm retracting it from the knowledge base. The period during while some norms are active are limited and conditioned to the addition of some facts in the knowledge base. The third and final rule points out the violations. Prohibitions are violated if facts are inserted into the knowledge base during while they are forbidden and permissions are violated if the facts are inserted into the knowledge outside the period during while they are permitted. The violations of obligations occur if facts are not inserted into the knowledge base in the corresponding period. The following Sections will demonstrate how to implement those rules according to the *temporal situations* and *if conditions* mentioned in Section 4.

### 5.3 Simple Obligations, Permissions and Prohibitions

Norms that describe obligations, permissions or prohibitions over the execution of actions without defining any temporal situation or if condition are always active. Such norms are never deactivated no matter what happens.

Although it is possible to describe obligations and permissions over the execution of a norm without stating any condition, it is not possible to state violations. Permissions characterize that such actions can always be executed, and, therefore, such norms are never violated. The obligations characterize that the actions must be executed but do not state when the executions must be checked. Thus, for each obligation or permission that is not associated with any temporal situation or if condition, only one rule that states the norm must be described.

On the other hand, prohibition can do be checked and violations can be fired in case the action is executed. Therefore, for each norm that describes prohibition for the execution of an action, two rules need to be defined: (i) to assert the prohibition; and (ii) to assert the violations if the forbidden facts are added to the knowledge base.

In order to exemplify the use of Jess we describe the implementation of *norm 5*. Rule (i) asserts the prohibition that is not conditioned to any fact. Rule (ii) asserts the violation if a kicker gets the ball.

```
(defrule forbidden:KickerGetBall          ;(rule i)
=> (assert (FORBIDDEN-non-dialogical-action-plan (entity kicker)(plan play)
                                                 (action getBall))))

(defrule violation:KickerGetBall          ;(rule ii)
?fact <- (non-dialogical-action-plan (entity kicker)(plan play)(action getBall))
?forbidden <- (FORBIDDEN-non-dialogical-action-plan (entity kicker)(plan play)
                                                    (action getBall))
=> (assert (VIOLATION (norm-violated (fact-id ?forbidden))
                      (action-done (fact-id ?fact)))))
```

### 5.4 Before the Occurrence of a Fact

Obligations for executing an action X before the occurrence of a fact W are verified testing if X has been executed before W occurs. For governing such norms three rules are defined: rule (i) asserts the obligation for execute X; rule (ii) retracts the obligation if X has been executed and W occurs; and rule (iii) asserts a violation if W occurs but X has not been executed (what can be verified by the existence of the obligation).

Permissions for executing an action X before the occurrence of W are verified testing if X is executed after W. In such case, the execution of X is not permitted. These norms are governed by three rules: rule (i) asserts the permission for execute X; rule (ii) retracts the permission if W occurs; and rule (iii) asserts a violation if W occurs and X is executed.

Prohibitions for executing an action X before the occurrence of an action W are verified testing if X is executed and W has not occurred. Such norms are also governed by three rules: rule (i) asserts the prohibition; rule (ii) retracts the prohibition if W occurs; and rule (iii) asserts a violation if X is executed and W has not occurred (what can be verified by the existence of the prohibition).

*Norm 1* is a good example for illustrate the implementation of norms that govern the actions that must be executed before another one. Since the norm defines that a referee is *obliged* to check the equipment of the players *before* starting the game, three rules was defined to govern such norm. Rule (i) stated the obligation. Rule (ii) retracts the obligation if the referee has checked the player equipment when the game starts. Rule (iii) asserts a violation if the game has been started and the obligation still holds informing that the referee has not checked the equipment. The obligation governs a non-dialogical action that must be executed after a dialogical action.

```
(defrule obliged:CheckEquipment           ;(rule i)
 =>(assert (OBLIGED-non-dialogical-action-plan (entity referee)(plan managingGame)
     (action checkEquipment)(attribs players)
     (condition "BEFORE UTTER(game; si;INFORM(;referee;; [;gameStart;;;;;;]))"))))

(defrule retract:CheckEquipment           ;(rule ii)
(non-dialogical-action-plan (entity referee)(plan managingGame)
                            (action checkEquipment)(attribs players))
(dialogical-action (scene game)(state si)(performative inform)(sRole referee)
                   (message "gameStart"))
?obliged <- (OBLIGED-non-dialogical-action-plan (ntity referee)
       (plan managingGame)(action checkEquipment)(attribs players)
       (condition "BEFORE UTTER(game; si;INFORM(;referee;; [;gameStart;;;;;;]))"))
=> (retract ?obliged))

(defrule violation:CheckEquipment         ;(rule iii)
?fact <- (dialogical-action (scene game)(state si)(performative inform)
                            (sRole referee)(message "gameStart"))
?obliged <- (OBLIGED-non-dialogical-action-plan (ntity referee)
       (plan managingGame)(action checkEquipment)(attribs players)
       (condition "BEFORE UTTER(game; si;INFORM(;referee;; [;gameStart;;;;;;]))"))
=> (assert (VIOLATION (norm-violated (fact-id ?obliged))
                   (action-done (fact-id ?fact)))))
```

### 5.5 After the Occurrence of W or If W occurs

Obligations for executing an action X after the occurrence of Y (or if Y occurs) cannot be governed since it is not possible to affirm that the execution of X will never occur after the execution of Y. It is not possible to state a rule that fires a violation for such norm since the action X can be executed anytime after Y has occurred. In order to govern such norms it is necessary to state any temporal situation limiting the time for the execution of X after Y has occurred. The temporal concept *between* should be used instead of *after* or *if* for governing such obligations. *Norms 4* and *6* are example of norms that should be implemented by using *between*, as depicted in Section 5.6.

Permissions for executing X after the occurrence of Y can be governed by two rules: rule (i) assert the permission if Y occurs; and rule (ii) asserts a violation if X is executed but Y has not occurred yet.

The governance of prohibitions for executing X after the occurrence of Y is the opposite of the governance of the related permission. Such governance is also characterized by two rules: rule (i) asserts the prohibition if Y occurs; and rule (ii) asserts a violation if X is executed after Y has occurred or if Y is true.

In order to exemplify a norm that use the *if condition* we refer to *norm 2*. This norm defines that the coach cannot execute an action that substitutes players if the number of substitutions is equal or greater than 3. The prohibition governs a non-dialogical action that is condition to the state of an object.

```
(defrule forbidden:PlayerSubstitution     ;(rule i)
(attribute-value (objectORagent team)(attribute substitutions)(value 3))
=> (assert (FORBIDDEN-non-dialogical-action-plan (role coach)(plan managingTeam)
               (action substitutePlayer)(attribs outPlayer,inPlayer,team)
               (contract-pre "team.coach=coach")
               (contract-post "team.substitutions=team.substitutions@pre+1,
                         team.playersInField->excludes(outPlayer),
                         team.playersInField->includes(inPlayer)") )))

(defrule violation:PlayerSubstitution     ;(rule ii)
?fact1 <- (non-dialogical-action-plan (role coach)(plan managingTeam)
                                      (action substitutePlayer))
?fact2 <- (attribute-value (objectORagent team)(attribute substitutions))
?forbiden <- (FORBIDDEN-non-dialogical-action-plan (role coach)(plan managingTeam)
                (action substitutePlayer)(attribs outPlayer,inPlayer,team)
                (contract-pre "team.coach=coach")
                (contract-post "team.substitutions = team.substitutions@pre+1,
                                team.playersInField->excludes(outPlayer),
                                team.playersInField->includes(inPlayer)"))
=> (if (>= (fact-slot-value ?fact 2) 3 ) then
       (assert (VIOLATION (action-done ?fact1  ?fact2)
                          (norm-violated ?forbidden))) ))
```

### 5.6 Between Y and W

A norm that states an obligation for executing an action X after the occurrence of Y and before the execution of W is governed by three rules: rule (i) asserts the obligation for execute X if Y occurs; rule (ii) retracts the obligation if X is executed and if W occurs; and rule (iii) asserts a violation if W occurs but X has not been executed.

The permission for executing X between the occurrence of Y and W is governed by the following four rules: rule (i) asserts the permission for execute X if Y occurs; rule (ii) retracts the permission if W occurs; rule (iii) asserts a violation if W occurs and X is executed; and rule (iv) asserts a violation if X is executed but Y has not occurred yet (i.e., if the permission for executing X has not been fired yet).

Prohibitions for executing X between the occurrence of Y and W are governed by three rules: rule (i) asserts the prohibition if Y occurs; rule (ii) retracts the prohibition if W occurs; and rule (iii) asserts a violation if X is executed, Y has occurred but W has not occurred, that is equal to say if X is executed and the prohibitions is still activated. Note that the rules that govern both prohibitions and permissions while using the temporal concept *between* are the combination of the rules used to govern such norms using the *after* and *before* temporal concepts.

The use of *between* can be exemplified by *norm 3*. It states that the player is forbidden to leave the field between the beginning and the end of the game. The norm defines a prohibition to execute a non-dialogical action limited by the execution of two dialogical actions. Rule (i) asserts the prohibition if the first dialogical action is executed, rule (ii) retracts the prohibition if the second dialogical action is executed and rule (iii) declares a violation if the non-dialogical action is executed during while it is being prohibited.

```
(defrule forbidden:LeaveFiled     ;(rule i)
(dialogical-action (scene game)(state si)(performative inform)(sRole referee)
                   (message "gameStart"))
 => (assert (FORBIDDEN-non-dialogical-action-plan (role player)(plan moving)
           (action leaveField)(contract-pre agent.position@pre=inField)
           (contract-post agent.position!=inField ))))

(defrule retract:LeaveFiled       ;(rule ii)
(dialogical-action (scene game)(state si)(performative inform)(sRole referee)
                   (message "gameStop"))
?forbidden <- (FORBIDDEN-non-dialogical-action-plan (role player)(plan moving)
                (action leaveField)(contract-pre agent.position@pre=inField)
                (contract-post agent.position!=inField ))
 => (retract ?forbidden))

(defrule violation:LeaveFiled     ;(rule iii)
(dialogical-action (scene game)(state si)(performative inform)(sRole referee)
                   (message "gameStart"))
?forbidden <- (FORBIDDEN-non-dialogical-action-plan (role player)(plan moving)
                 (action leaveField)(contract-pre agent.position@pre=inField)
                 (contract-post agent.position!=inField ))
?fact <- (non-dialogical-action-plan (role player)(plan moving)(action leaveField)
            (contract-pre agent.position@pre=inField)
            (contract-post agent.position!=inField ))
=> (assert (VIOLATION (norm-violated (fact-id ?forbidden))
                   (action-done (fact-id ?fact)))))
```

Sections 5.3 and 5.5 point out that some obligations over the execution of a norm that cannot be governed. Since obligations need not to be fulfilled immediately after they were declared, it is necessary to inform the period during while the agents are being obligated to execute the action in order to govern them. *Norms 6* and *4* are very good examples of such obligations. *Norm 6*, for instance, defines that the referee must declare a penalty if a kicker gets the ball. However, this norm does not define how much time does the referee has to fulfill its obligation. Therefore, it is not possible to affirm that the obligation was not fulfilled since it can be at any time. In order to properly regulate such norm it is needed to provide a limit till when this obligation

must be fulfilled. *Norms 6* was adapted to inform that the referee has 1 minute to declare the penalty after the kicker has gotten the ball.

```
OBLIGED ( UTTER(game; si; PENALTY(;referee;kickerTeam;[;penalty;;;;;;ballTouch]))
  BETWEEN (kicker EXECUTE play:getBall, 1 MINUTES OF kicker EXECUTE play:getBall))
```

## 6 Conclusion

This paper proposes the implementation of norms[3] that govern dialogical and non-dialogical actions by using Jess. Our normative language makes possible the specification of non-dialogical norms that govern actions not related to messages being sent or received. As illustrated by the example, the specification of those kinds of norms is very important for governing multi-agent systems. In addition, we have also presented how to implement in Jess the rules that regulate several possible norms taking into account the three *deontic* concepts, the proposed temporal situations and if conditions. Our proposal was designed to receive information about the executed actions and observable attributes and to activate norms or assert violations of norms.

Although the current version does not contemplate sanctions and awards, it can be easily extended in order to do so. The sanctions should be provided when the related violations are fired. The awards should be supplied when the norms are retracted and no violation of such norms has been fired.

## References

1. Boella, G.; van der Torre, L.: Regulative and Constitutive Norms in Normative Multi-Agent Systems. In Proceeding of 9th Int. Conference on the Principles of Knowledge Representation and Reasoning. California (2004).
2. Artikis, A., Kamara, L., Pitt, J., Sergot, M.: A Protocol for Resource Sharing in Norm-Governed Ad Hoc Networks. Volume 3476 of LNCS. Springer-Verlag (2005)
3. Broersen, J., Dignum, F., Dignum, V. and Meyer, J. Designing a deontic logic of deadlines. In 7th Int. Workshop of Deontic Logic in Computer Science (2004)43-56.
4. Cranefield, S.: A Rule Language for Modelling and Monitoring Social Expectations in Multi-Agent Systems. Technical Report 2005/01, Univ. of Otago (2005)
5. Dignum, F., Broersen, J., Dignum, V., and Meyer, J. Meeting the deadline: Why, when and how. In 3rd Int. Workshop on Formal Approaches to Agent-Based Systems, (2004)30-40.
6. García-Camino, A., Rodriguez-Aguilar, J, Sierra, C, Vasconcelos, W. Norm-Oriented Programming of Electronic Institutions: A Rule-based. In Proc of the 5th Int. Joint Conf. on Autonomous Agents and Multiagent Systems, ACM Press (2006) 670-672
7. García-Camino, A., Noriega, P., Rodríguez-Aguilar, J.A.: Implementing Norms in Electronic Institutions. In: Proc. of the 4th Int. Joint Conf. on Autonomous Agents and Multi Agent Systems AAMAS'04, ACM Press (2005) 667-673.
8. López, F.: Social Power and Norms: Impact on agent behavior. PhD thesis, Univ. of Southampton (2003)

[3] The full normative language described in the paper and the Jess program used to illustrate our approach are available at http://maude.sip.ucm.es/~viviane/products.html.

9. López, F, Luck, M. and d'Inverno,M. Constraining autonomy through norms. In Proc. of the 1st Int Joint Conf. on Autonomous Agents and Multi Agent Systems(2002) 674-681
10. Meyer, B. Object-Oriented Software Construction Prentice Hall, second edition (1997)
11. Singh, M.: An Ontology for Commitments in Multiagent Systems: Toward a Unification of Normative Concepts. Artificial Inteligente and Law v. 7 (1) (1999) 97-113.
12. Vázquez-Salceda, J., Aldewereld, H., Dignum, F.: Implementing Norms in Multiagent Systems. LNAI 3187. Springer-Verlag (2004) 313 - 327