Regulated MAS: Social Perspective

Pablo Noriega[1], Amit K. Chopra[2], Nicoletta Fornara[3], Henrique Lopes Cardoso[4], and Munindar P. Singh[5]

1 **IIIA-CSIC, Spain**
2 **Lancaster University, UK**
3 **Università della Svizzera italiana, Switzerland**
4 **Universidade do Porto, Portugal**
5 **North Carolina State University, USA**

Abstract

This chapter addresses the problem of building normative multi-agent systems in terms of regulatory mechanisms. It describes a static conceptual model through which one can specify normative multi-agent systems along with a dynamic model to capture their operation and evolution. The chapter proposes a typology of applications and presents some open problems. In the last section, the authors express their individual views on these matters.

1998 ACM Subject Classification I.2.11 Distributed Artificial Intelligence

Keywords and phrases NormMAS, Norms, Open Interaction

Digital Object Identifier 10.4230/DFU.Vol4.12111.93

1 Introduction

Central to the idea of a normative multi-agent systems is the important distinction between regimentation and regulation, first drawn in the present technical setting by Jones and Sergot [42]. The key distinction is that regimentation arises in a system that forces or precludes certain actions whereas regulation arises in a system that neither forces nor precludes the relevant actions but merely regulates the participants so that those actions respectively occur or fail to occur. Specifically, when we think of multi-agent systems—which by definition consist of multiple agents—subject to various constraints, regimentation becomes ensuring the agents are simply (i.e., "physically") unable to violate some constraints whereas regulation becomes ensuring the agents choose not to violate the constraints, despite being able to violate them.

In other words, the notion of regulation is central to the idea of autonomy and thus legitimises the general idea of norms as we understand them. For if an agent were simply unable to violate a constraint, the constraint would appear less like a norm and more like a physical law, such as that of the conservation of energy.

A note on terminology: for our present purposes, we treat a regulation as a norm that is of social provenance and applies on the interactions of the participants. In this manner, we would not include within regulations the following kinds of norms: personal norms (never mislead my friends) and social conventions (greet everyone at the start of a meeting).

Regulation as the idea of control despite autonomy involves the obvious idea of a life cycle of regulations being promulgated, obeyed (or disobeyed), enforced, updated, and revoked. The notion of regulations presumes what we term a *normative architecture* or a social backdrop. The notion of regulations, however, is more general than any such normative architecture in which they may exist—regulations can exist in any setting where multi-agent systems make sense. Specifically, we can see regulations being applied in a setting involving a designated

Normative Multi-Agent Systems. *Dagstuhl Follow-Ups, Vol. 4.* ISBN 978-3-939897-51-4.
Editors: Giulia Andrighetto, Guido Governatori, Pablo Noriega, and Leendert W.N. van der Torre; pp. 93–133
DAGSTUHL FOLLOW-UPS Dagstuhl Publishing
Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Germany

governor [54] for each agent, just as much as in a setting involving social control. And, regulations can be promulgated by a central authority just as much as being democratically decided. Our emphasis, as multi-agent systems researchers, falls on normative architectures that respect the autonomy of the participants and carry a conceptually decentralised flavour. Even such architectures may be realised in systems wherein the participants elect a governor who either enforces the regulations they jointly promulgate or promulgates and enforces regulations on its own initiative. It is not surprising that many computational architectures reflect one of the cases where a governor is either elected or appointed, even if the remaining participants might be notionally peers of each other.

1.1 Example

To motivate this discussion further, let us consider the example setting of traditional commerce. Commerce clearly involves autonomous parties: buyers and sellers at the very least and often members of an extensive ecosystem of suppliers, shippers, payment processors, and ratings agencies. Since the parties are autonomous, regimentation is out of the question: you cannot prevent a seller from selling illegal goods or from failing to ship goods that the buyer has paid for nor the buyer from refusing to pay for goods ordered. But regulation is essential. The buyer and seller can regulate their transaction to some extent by entering into a contract that specifies the transaction. Or, they may adopt the regulations in place in their social environment, such as the city or marketplace where they operate. In either case, in general, each party would rely upon another entity to help enforce a regulation when its interests are at stake in the satisfaction of the regulation. This entity could be a government agency, an industry board, or a nominated third-party that the participants agree upon. Although the parties may also function without such an entity to back the regulations, such a situation becomes rarer as the stakes go up.

Now what would change if we move to electronic commerce? Clearly, the same or similar roles are still involved. It is obvious that there is an equal need for regulation in virtual settings as well. E-commerce is usually facilitated through marketplaces such as eBay wherein the buyers and sellers can meet to conduct business. The marketplace serves as a promulgator and enforcer of regulations. Thus the most common form of e-commerce employs a centralised architecture. In the days of Usenet prior to the Web, it was common for users to find each other and conduct transactions without a formal marketplace. This is analogous to transactions where the parties find each other through Craigslist today. Such transactions might involve one party sending a payment to another to purchase the specified goods. In such settings, too, regulations apply though there is no ready means to enforce them. Potentially, in some countries such as in the US and Europe, the government can get involved if broader regulations against fraud in commerce appear to be violated.

Let us imagine that we have a virtual marketplace. Clearly the entities that live in such a marketplace are not people but their software agents. This leads to the question of to whom do the regulations apply: the agents or the people? An important idea here is one of responsibility and accountability, for example, as delineated by Mamdani and Pitt [50]. One can imagine a virtual world populated by software agents where each agent acts on its own behalf and is therefore subject to the virtual world's regulations. But in the more common uses of agents, especially in settings such as commerce that involve an external reality, the agents are merely representatives of people. The agents could be intelligent and function without minor guidance but insofar as they are representatives of people, the regulations apply to people, who must bear the consequences of obeying or disobeying them. The situation is analogous to a business owner using an accountant to prepare a government filing. It is the business owner who is subject to government regulations and would bear the consequences of complying or not, unless the accountant has violated regulations pertaining to the field of accountancy.

Restricted in this manner, we view each agent as representing a principal. The computational system, such as the marketplace above, facilitates interactions among the agents by supporting the necessary bookkeeping but does not have a life of its own. That is, the computational system is merely an instrument. The life belongs to the participants, including the principal whom the marketplace viewed as an agent represents. Actions in the computational systems *count as* actions in the real world where they are subject to appropriate regulations.

1.2 Layout of the Chapter

The main objective of this chapter is to formulate the research challenges of regulated MAS. The remainder of this chapter is organised as follows. Section 2 takes a closer look at some typical applications of regulated MAS. Section 3 provides a deeper motivation of normative architectures wherein regulations are feasible, discussing the common traits of normative architectures. Section 4 describes the main components of a conceptual model for regulations. Section 5 introduces how such a conceptual model can be operationalised in the above architectures. Section 6 discusses some dynamic aspects of regulations, especially how they evolve over time. Section 7 relates regulated MAS with other perspectives in computer science, with an emphasis on the software engineering of sociotechnical systems. Section 8 illustrates a real-life example of a regulated MAS drawn from the domain of open innovation. Section 9 summarises some open research problems pertaining to regulated MAS. Section 10 provides a soapbox for each of the authors to describe their personal views.

2 Normative Applications

Regulated MAS serve one main function, to set a "level playing ground" (as D. North [55] postulates for institutions). Putting it bluntly, regulated MAS create an *institutional reality* that is different from the physical reality. In the institutional reality, only *institutional facts and actions* exist. As we discuss in the next section, these two realities have some correspondence thanks to the "constitutive" norms. Those norms produce, on one hand, the legitimacy of the regulated MAS to create an institutional reality that is governed by regulations and to enforce these on participants and, on the other hand, the entitlements needed by agents to act within that institutional reality and consequently held liable in the actual world. In addition, those constitutive norms also determine the ontology that will exist in the institutional reality and the *counts-as* relationship that establishes a correspondence between facts and actions in the physical world, and institutional facts and actions (see Searle [62]). The ultimate purpose of a regulated MAS is to articulate interactions where several *autonomous* agents may be involved. By fixing ontology and regulating admissible and legal actions, the regulated MAS reduces uncertainty and simplifies decision-making "surrounding agents with reliable and perceivable patterns of events that allow them to make reasonable and stable calculations about behaviour" [65, p. 78]. Furthermore, governance, by providing some degree of control over undesired behaviour, serves to allocate risk and limit the exposure and liability of agents.

A regulated MAS is supposed to be implemented *properly* (with respect to the three integrity challenges discussed on Page 99), and to make the aforementioned institutional reality work. The implementation also needs to support five components that are essential for that *normative architecture* we mentioned above:

- A “virtual space” where agents may interact.
- A “shared ontology” to which all agents may univocally refer to.
- An “interaction model” that determines what the primitive or atomic agent actions are and how they may be interlaced into activities involving many agents.
- A “set of regulations” that affect agent interactions. In addition a collection of norms of different sorts, this set may contain an explicit mention to regimented constraints and may also include (*meta*)norms that regulate how existing norms may be modified or revoked and how new norms may be introduced in the set.
- A “governance model” which consists of two complementary elements. First, some principles about compliance (i.e., constraints—within the regulated MAS—whose enforcement is regimented, norms whose enforcement is discretional—in the sense that the sanctions may be imposed or not depending on the judgement of the enforcers—and yet another type of norms whose application is not discretional). Second, enforcement mechanisms that contend with violations; that is, how infractions are ascertained and then dealt with (e.g., a centralised mechanism, some law enforcement agents, self-regulated peer-to-peer control).

What are the conditions that make a regulated MAS useful in practice? There is a threefold answer:

First, when the agents whose participation is regulated have the following features: (i) they are autonomous towards observance of norms (ii) their internal decision-making and motivation is beyond the control of the regulated MAS, (iii) the agents may be malicious or incompetent and thus are likely to infringe norms, (iv) may “enter or leave” (be active in) the virtual space at will, (v) are independent from each other or do not represent the same principal and thus may have different and perhaps conflicting individual goals, and (vi) there is some liability in their actions.

Second, when the situations where these agents meet have the following features: (i) regulated activities are repetitive (no need to regulate one-shot situations), (ii) activities are performed in a shared social context (at any point in time several individuals share the same state of the world, the same regulations apply to all and any institutional action by any of them affects the shared state of the world) (iii) regulated interactions are perceivable and applicable conventions are ostensible.

Third, the regulated MAS is backed by an organisation and supported by technological artifacts that resolve the “integrity challenges” that are discussed in Section 5. Namely: (i) create the stable interaction environment and maintain it in proper operation, (ii) manage access and “identity” of participating agents, (iii) implement the governance model, (iv) assure the persistence and enforcement of regimented constraints as well as the sound management of regulations.

2.1 Towards a typology of regulated MAS applications

Since the notion of a normative MAS is recent and few multi-agent system approaches include normative aspects explicitly as part of their conceptual framework there are not many examples of actual regulated MAS applications. Nevertheless, the examples reported in the last chapter of this book provide a suggestive indication of the types of applications of regulated MAS that are being or will be developed. Other examples of applications, which may qualify as regulated MAS in the sense just described, may be drawn from a number of "standard" MAS that have been developed from an organisational or an institutional standpoint. Additionally, some conventional systems and even some multi-agent systems that organise social or collective activities may be reified as regulated MAS, even if the actual normative component in some of these maybe flimsy. Note, however, that for all these examples, the provisos stated in Section 7 should be kept in mind.

A look at those three sources of examples allows us to venture four distinct types of applications that involve agents, situations and organisational functionalities for which regulated MAS are appropriate.

Hard-wired sociotechnical systems. Systems where the conventions that regulate agent interactions are established at design time and are issued and maintained by the owner of the system [69]. Although many of these conventions are hard-wired into mostly static workflows in a regimented way, there may also be some norms that might be violated and need enforcement and regulations may evolve over time. The balance between regimentation and enforceability (and the corresponding enforcement mechanisms as well as the dynamics of the regulations) respond to pragmatic aspects like efficiency, ease of use, accountability, trust-worthiness, risk, and liability. Typical examples are e-markets (for on-line sales, e.g., eBay and PayPal), enterprise information systems (for hospital management, hotels) and web-based conventional social activities (for example, e-government transactions, e-learning or some forms of mobile health care).

Agent-reified conventional regulated environments. This type includes social conventional or traditional activities that are subject to norms, but are now web-enabled in some way and involve agents that perform regulatory functions; for example, collective writing in Wikipedia. Some of these sociotechnical systems may be properly labeled regulated MAS. Examples of these type in Chapter 7 of this book are the one about norms in open source software repositories by Savarimuthu and Dam; the one by Villata and Gandon, on data licensing in the web; and the one by Fornara and Eynard on data collection.

Artificial social systems. Two distinct breeds: (i) Those systems used for modelling and simulation in fields like experimental economics, sociology or policy-making; for example, the UAV (unmanned autonomous vehicles) example by Governatori and Lam, and the water management policy simulator example by Noriega (both in Ch. 7) and (ii) Virtual worlds, for immersive remote interactions, like the ones discussed by Cranefield and Verhagen (in Ch. 7); and virtual worlds as those used in games, like those discussed by Dignum (also in Ch. 7). In both types of applications, the advantage of a regulated MAS is that design concerns about agents and the environment are clearly separated; the use of explicit formal norms allows for a more abstract and flexible specification of conventions, scenarios and performance indicators, and in addition these may support to some extent off-line and on-line reasoning by designers and agents.

Open sociotechnical ecosystems. In these systems a regulated MAS furnishes the normative architecture that enables the inception and runtime support of new organisations, agreements or collective activities that are subject to their own regulations and may be created by centralised off-line design or by on-line peer-to-peer collaboration of participating agents. Examples of these systems are the virtual enterprises scenario proposed in [45], the Ocean Observatory Initiative (OOI) reported by Singh et al. in Chapter 7 of this book and the Green Open Innovation Platform described below in Section 8.

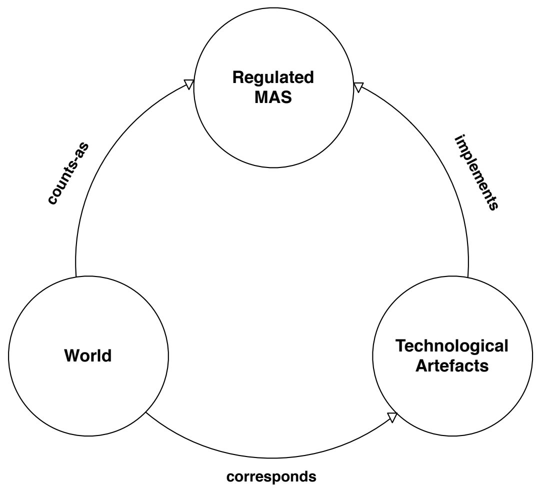

Figure 1 The relationships between the parts of a sociotechnical system.

3 Normative Architecture

The foregoing sections made the case for regulated MAS as involving agents who represent principals. Let us now consider the architecture of a regulated MAS, which we dub the *normative architecture*, in conceptual terms. The agents function as autonomous with respect to other agents but they derive their autonomy from the principals they represent, thereby making the principals accountable for their actions. In this manner, regulations make the most sense in settings that combine the social and the technical spheres. In this regard, we can treat regulated MAS as *sociotechnical systems* [69]. Sometimes this term is used for systems that merely involve human interaction with a computer; here, we specifically mean interactions between socially autonomous entities, such as people and organisations, though the interactions involve and are mediated through technical artefacts such as computers.

Figure 1 describes how a regulated MAS may be understood. At one side is the world, which we can think of approximately as the "physical" world. More carefully, it represents the objective reality for the purposes of modelling. For example, in commerce, the world provides a home for the goods being bought and sold as well as the infrastructure such as for shipping and paying that commerce needs. In a multi-user virtual environment, the world may simply be the virtual reality in which avatars of the users exist and function.

At the other side lie the technical artefacts. For our purposes, these are computational representations of the world as well as means (programs) to manipulate them. We can resolve the distinction between representation and reality simply in terms of how we choose to model. Whatever is endogenous falls into the technical artefacts and whatever is exogenous falls into the world. For example, although physical goods, vehicles, currency are clearly the province of the world, we might treat the services that deal with such objects as internal or endogenous. A key distinction is that whatever is endogenous can be manipulated computationally from the model and vice versa. Referring back to Section 1, the endogenous parts can be regimented (the bank account is never overdrawn) but what is exogenous can only be regulated (the customer shouldn't write a check for more money than he has in his account) [69].

At the top, we place the regulated MAS itself, which exists in the world and is realised through the technical artefacts. The whole idea of commerce, for example, resides at this level. People may pass objects back and forth but only in some suitable settings can such

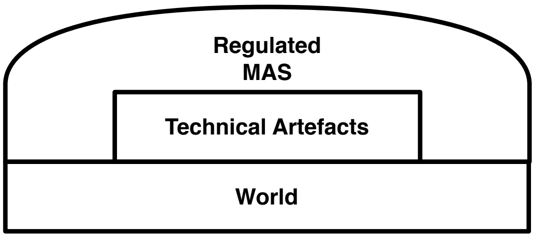

Figure 2 A sociotechnical system schematically.

actions *count as* [62] a commercial transaction. In particular, these settings involve the satisfaction of various norms, and impose regulations upon the transaction—for example, that goods can only be sold by someone who has ownership of them (not just possession) and a successful transaction results in a change of ownership.

The above characterises a regulated MAS in conceptual terms. How can we realise such a system? Figure 2 describes a sociotechnical system in schematic terms. We view such a system as having three main parts. The sociotechnical system is grounded in the physical world, inhabited by exogenous resources such as vehicles (autonomous or otherwise). Overlaid on the physical world is the technical (i.e., computational) structure, inhabited by information resources such as databases. Normative reality, home to the regulations, exists over the physical world and technical artifacts. In this manner, social interactions—that is, those subject to regulations—may be realised in the physical world through the mediation as needed of the technical artifacts.

The main conceptual challenge in regulated MAS is to maintain their *integrity* if only to ensure that regulations are being obeyed or to find out whether they are being disobeyed and by whom. Ensuring integrity in a sociotechnical system is nontrivial. Since a sociotechnical system brings together three concerns in its modelling, it is important to recognise that each of them may potentially be the cause of an integrity violation. First is the challenge in ensuring that the regulated MAS itself is sound: that is, the regulations do not conflict with each other and are potentially jointly achievable in principle. Chapter 2 addresses these formal challenges.

Second is the challenge of ensuring that the technical artefacts are correct. Let us put aside implementation-level concerns, such as that the sorting routine being used is correct, which we can capture in terms of the underlying infrastructure. The main conceptual aspect of integrity of the artefacts comes down to what is termed *identity* in computing parlance [20] but better corresponds to an ability to identify relevant objects. Identification is a crucial function of sociotechnical systems. Not surprisingly, the ability to identify principals is crucial for accountability: if we don't know who's who, we have no legitimate basis for determining if a regulation is being obeyed or disobeyed. The ability to identify technical artefacts is equally as crucial because it enables tracing actions to their principals and thus potentially determining who was responsible for obeying or disobeying a regulation. In traditional human societies, for example, a small village, each party would be known to every other party. In larger human societies, identification becomes difficult. In modern settings, the government, which plays the "governor" role alluded to in Section 1, additionally provides a means to identify the participants. Likewise, in virtual settings such as eBay, the resident authority (think of it as the governor of the eBay marketplace) provides the identification.

The provisioning of identification is a common feature of regulated settings. An ability to identify a participant is a prerequisite for regulating its interactions. Identifiability can be achieved definitionally at the regulated MAS level. It can be trivially implemented at the technical level: simply give everyone an account with which they must login before participating in the sociotechnical system. Achieving identifiability at the physical level is nontrivial and the challenge segues into the one below.

Third is the challenge in ensuring that the physical world, which recall provides the infrastructure upon which actions and events of relevance to the regulations arise, is not corrupted. For example, if the underlying messaging system is corrupted and delivers bogus messages or fails to deliver correct messages, the integrity of the regulated MAS would be violated. In general, such threats from the infrastructure are a major security threat to a regulated MAS. Figure 2 captures the intuition that the regulated MAS ought to control the relevant aspects of the physical world. For example, eBay viewed as a regulated MAS controls the infrastructure through which auctions are created by sellers and bid submitted by buyers. Such control is essential for eBay to determine which bids were in time, who if anyone won an auction, what item did they buy, and how much they committed to pay for it. In essence, the regulated MAS takes responsibility for the physical world as a way to ensure its own integrity. The above assumption of control, however, may not hold in practice—and arguably is never met in practice. For example, underlying eBay's infrastructure and its execution by users lie computers and networks that eBay does not control. Thus we would need techniques for regulated MAS to function correctly despite threats from the infrastructure. Indeed, human societies are not paralysed because of the existence of such threats and norms can provide a way of dealing with them.

4 Conceptual Model

This section describes the fundamental application-independent components that all open interaction systems have in common. We base the foregoing claim, on our experience [19, 25, 45]. Those application-independent parts should be integrated with application-dependent concepts and concrete values of some parameters in order to realise an actual interaction system. The main advantage of this model is that, in principle, it may be used for the specification of a different type of systems used in different domains, from the definition of marketplaces for the improvement of e-commerce to the definition of collaborative/coordinating/social ecosystems for supporting collaborative work and social coordination. In the following we will describe what we consider are the main components for the design of those systems, how those components should be used for the realisation of a system, and the main functionalities that should be implemented in those systems for their correct execution. A fundamental assumption behind the definition of this model is that the interacting parties are *autonomous entities*; that is. these agents may be human or software, each agent has its own goals, each one belongs to a specific principal and, furthermore, each of these agents may violate the norms that regulate its interactions with others.

4.1 Design Components

We may distinguish between those components whose main goal is to *enable social interaction* among the participants agents, and those components whose main goal is to *regulate such interactions*. Regarding the first type of components, we need to specify a set of *conventions* for realising conversations or interactions. Those conventions regard:

- The definition of the common *ontologies* that the agents need to share in order to interact. They consists of concepts and properties used for describing the objects on which the interaction is focused and the shared knowledge of the interacting parties that evolves during the interaction. Those ontologies have an application-independent part that defines for instance the concept of action, event, obligation, and so on, and an application-dependent part focused on the domain of the application. Some of the concepts represented in those ontologies have a direct mapping to objects in the real world, some other exists only in the institutional reality of the regulated MAS.

- The definition of those *actions* that agents may perform. We assume that within the regulated MAS, agents use only *communicative acts* for interacting. Those acts can be defined in terms of the *preconditions* for their successful performance and in terms of their *effects* on the state of the interaction, that is, on the state of a set of application independent and dependent components whose existence is jointly accepted by the interacting agents. For example, the effects of some communicative acts may be to create or modify the *normative relationship* among the agents: an agent that bids for a product in an auction commits to pay the amount of the bid. The fact that by performing a communicative act an agent can change normative relationships at runtime, can be modelled using constitutive rules [62] (X counts-as Y in C) for associating an appropriate semantics to the communicative acts that agents perform within a specific protocol enactment, except for declarations [27].
- The definition of the relevant *events* that may happen during an interaction, (again in term of preconditions and effects), the most common of which is the passing of time. Events may be referred to in the content of communicative acts, for example, as a promise to perform an action before a given deadline.

For example, in the definition of a marketplace the domain language might contain a definition of the different types of products that may be exchanged, their properties, some constrains that specify the conditions under which the values of their properties are correct—for example rules for changing the price of a product—and some terminological properties that are valid for the domain in question—like the fact that a bank transfer is a particular type of payment. The actions can be the act of buying, selling, paying, and delivering a product.

Regarding the components of the conceptual model devoted to regulate the interaction or conversation of the agents, it is important to observe that very often in human social life, interactions happen in a specific *context*; for example in a school, in a bar, in a market, in a university. The context is useful for disambiguating the meaning of certain terms and their properties, and for the fact that it introduces some predefined normative relationship among the agents playing certain roles. This is usually done with the aim of bringing the interaction to a certain final goal, for example an auction can be used to reach an agreement on the price of a given product, an exam is used to grade a student. Moreover, this is useful for avoiding the complex task of starting every negotiation from scratch, where agents need to reach agreements on the rules of the interaction.

As the preceding remark suggests, it is fundamental to clearly define the *context* where the interaction will take place and the *norms* that will regulate the interaction in that context. It is also important to regulate agent interactions in such a way that agents may benefit from their autonomy. In order to make it possible for the interacting agents to profit their autonomy, it is important to regulate their interaction in a way that allows them as much freedom as possible to choose what communicative action to perform among the set of available ones. Therefore it is necessary to be able to express a list of normative constrains that can specify:

- What actions are permitted. One possible default for the system is that if an action is not permitted, it is prohibited, and the prohibition can be violated. Another default may be that every action by default is permitted, unless it is explicitly prohibited, and the prohibition can be violated.

- What actions are prohibited in a regimented fashion (prohibitions that cannot be violated). In the OCeAN model [25] in order to avoid confusion between this notion of prohibition and the previous one, this notion of prohibition is formalised using the notion of *institutional power* [43]: if an agent does not have the institutional power to perform an institutional action (an action that changes the value of a property whose value is shared by all the agents involved in the interaction, the effects of the attempt to perform the action are void;
- What actions are obligatory, and obligations can be violated. Very often an obligation has an associated deadline that specifies the instant of time when the obliged action has to be performed.

Given that those context-dependent norms are often defined at design time, they are expressed in terms of the *roles* that the agents will be able to play at run time. Those roles very often are related by subsumption and incompatibility relationships that create a social model for the agents.

Usually those normative constraints are in force in a specific and delimited context of interaction that should be clearly represented in the model and should be distinguished from other contexts where usually other norms, other roles, and other ontologies apply. Those contexts of interaction may be called *scenes* [54], *spaces* [73], *orgs* [69], or groups (when their distinguishing characteristic is represented by the agents involved in the interaction). The introduction of the construct of contexts of interaction as an application independent component of the model requires to define the rules for regulating the creation at runtime of new contexts of interaction, for letting agents enter and exit from contexts, and for destroying a given context.

4.2 Design Functionalities

In order to actually enable an open interaction among autonomous-heterogenous agents belonging to different principals, the previously described components should be properly formalised for the specific application domain where the interaction system will be used. For example, in the definition of an e-marketplace the different types of auctions (English, Dutch, double, ...) and contracts should be specified (these are the contexts of interaction), in terms of domain ontologies, roles (seller, buyer, auctioneer, participant, ...), actions (buying, selling, paying, bidding, ...), and norms for regulating the performance of the actions available in a given space.

Once a system is defined, it is necessary, before its execution, to test if the formalisation has some specific properties. For example, to test that the definitions of all the ontologies are consistent, that the defined roles are all used, that the preconditions for the performance of the intended actions may be satisfied and that that the applicable norms are not in contradiction—in spite of the fact that when norms refer to intervals and instants of time, deadlocks are difficult to check at design time and hence usually need to be dealt with at runtime. Another important functionality is the one for proving that all the possible evolutions of the interaction system are constrained within a given set of boundaries that will allow the interacting agents to reach certain predefined goals.

Usually the process of realising those functionalities depends on the language used for formalising the system at design time (see 9.1.1 below). For example, if logical languages like decidable description logics (DL) (that are the basis of Semantic Web Languages like OWL) are used, then some of those properties may be checked thanks to the use of available DL reasoners (like Pellet or HermiT[1]).

Once a formal specification is finalised and some of its properties are checked, it will be used at runtime for actually executing an open dynamic interaction system whose components, i.e., agents and norms, may dynamically change during the interaction. This process will be described in detail in the following section.

5 Operational Model

Given the conceptual model introduced in the previous section, here we describe the challenges that are important to take into account from an operational perspective. That is, we identify the specific concerns that need to be addressed when developing a normative multi-agent system.

5.1 Operational Components

The first challenge one needs to address when designing a normative multi-agent system is *promulgation*: when and how are norms created into the normative environment within which norm-regulated interactions take place. There are (at least) two possible approaches.

Design-time norms. The most common approach regarding norm promulgation consists of relying on the system designer to anticipate the possible encounters that are to be governed using a normative approach, and to define the norms most suitable to those encounters. While keeping the possibility of addressing an open scenario in which interacting agents are concerned, this approach assumes that norm promulgation is a design challenge. Such perspective fits many real-world applications identified in Section 2.

Runtime norms. Some applications of normative multi-agent systems inherently require that the norms applicable to a certain interaction are at least adopted at runtime. A typical case is electronic contracting [45]: when establishing a contract, two (or more) agents negotiating on behalf of their principals choose the normative setting that will regulate the enactment of such contract (e.g., by specifying the type of the contract they are establishing). More sophisticated norm specification mechanisms include, on one side, the negotiation (or configuration) of norms, as well as their assembly and, on the other side, the emergence of social norms as discussed in Chapter 5.

Another operational concern in a normative multi-agent system is related to the *observability* of agent actions. This concern is directly related with the functional purpose of a norm, which is to be able to distinguish compliant from noncompliant behaviours. It is therefore a crucial aspect of norm *monitoring* to be able to assess the relevant agent actions that enable the normative environment to determine whether norms are being complied with or not. Observability is particularly tricky when dealing with prohibition norms. In some applications (e.g., auditing), when dealing with obligations an assumption of self-motivated compliance demonstration can be in place, i.e., it is in the best interest of agents to publicise fulfilment. Detecting prohibition violations, on the other hand, is much harder because it demands for a pervasive character of the monitoring system; in practice, different verifiability levels exist [76] regarding the actions agents can perform.

[1] http://clarkparsia.com/pellet/, http://hermit-reasoner.com/

Closely related with this challenge is the choice of implementation of norms [38], which is tightly coupled with the freedom agents are allowed when playing in the normative environment. Two approaches have been pursued: (i) *regimentation*, which prevents unwanted outcomes by imposing constraints; (ii) *enforcement* [37], which consists of using mechanisms to influence the decisions agents make, letting them choose whether to fulfil or violate norms.

Regimented norms are only viable in totally controlled scenarios, in the sense that agent autonomy is reduced by making sure that disallowed behaviours do not have any effect in the system (this is the original approach to engineering electronic institutions [54, 21]). On the other hand, enforcement means that external (as far as the agents themselves are concerned) mechanisms must be put in place so as to influence agent behaviour. Examples of such mechanisms include, among others, sanctions (either normative or utilitarian), incentives, or reputation.

Sanctions, in particular, may be employed following two different (nonexclusive) policies: retribution aims at compensating the addressee of a violation, while deterrence puts an emphasis on punishing the violator so as to discourage future violations. Taking into account this distinction, it may be desirable to incorporate both sanctions that concern actions to be performed by the violator and sanctions that materialise as actions that the norm enforcer is authorised to perform [24].

As stated above, monitoring is a central operational component in normative multi-agent systems. In rich social interaction scenarios there may also be a need to address *blame assignment*, in the sense that norm infraction may not entail guilt of the agent that is the subject of the violation taking place.

Summarising, when designing a normative multi-agent system a number of core operational components have to be engineered. This section has identified norm promulgation, observability, monitoring and enforcement as the key elements to be addressed. Additional concerns are raised in Section 9.

5.2 Operational Functionalities

From a functional perspective, and from the point of view of interacting agents, when designing a normative multi-agent system, a number of concerns need to be addressed. These are related to the operational components identified above.

The first functionality is important in open systems, and concerns the possibility for agents to enter and leave a normative multi-agent system. When entering the system, an agent adheres to the set of norms that regulate agents behaviour. Admission may encompass specific constraints that must be met. Leaving the system may demand for checking certain conditions (e.g., the agent must not have any pending obligations).

Another important functional aspect is the establishment and updating of norms at runtime. When enabling this possibility, it is necessary to engineer appropriate mechanisms that allow agents to negotiate the norms that guide their further interactions and to feed those norms into the normative environment. Accommodating specific infrastructures may facilitate this task, e.g., through coordination artefacts [40] or normative frameworks [46].

Naturally, the next functionality concerns enactment, i.e., designing the means through which agent interactions are assessed by the system. This is intrinsically related with the challenge of observability, and is crucial to allow for monitoring to take place.

6 Evolution

In general terms, when considering the evolution of a normative system, one needs to consider two kinds of changes: norm *promulgation*, which consists on establishing a new norm; and norm *derogation*, which removes a norm. Modifying a norm, e.g., by changing its applicability conditions, may be seen as a derogation followed by a promulgation.

A first step towards encompassing evolution in the normative multi-agent system has already been identified in the previous section, and concerns normative dynamism from the agents' point of view. When norms are to be created at runtime [45], the normative environment denotes an evolving facet. It may also be the case that within a particular institution, different organisations are established at runtime. Those organisations may have their own normative structure [75].

Typical in these approaches is to embed into the normative system some kind of infrastructure determining the normative changes that can be introduced. This infrastructure, specified at design-time, dictates the changes at runtime that agents may introduce in the normative system (their *degrees of freedom* [1]).

Dynamism may also be an important and desirable property from the normative system's point of view. In this case, two questions need to be addressed. The first question relates to how to evaluate the performance of the system as a whole. The second one relates to the changes that can be introduced in the normative multi-agent system so as to improve its overall performance.

Performance evaluation demands for an analysis of how effective a normative system is in terms of regulating the multi-agent environment. One may approach this issue by observing the behaviour of agents and assessing if the population as a whole complies with the norms: if it does not, some changes in the normative system may need to be introduced, e.g., by adjusting enforcement levels (as in [49]). In some cases, however, excessive or inadequate regulation may hinder the system by preventing some better overall performance from taking place. In such situations, although agents may be complying with norms, it might be in the best interest of the system for them to do otherwise.

Noncompliance can therefore be seen as an opportunity to change the normative system, by considering violations as alternative enactments that must be reacted-upon through changes in the normative structure. Those changes are not targeted to deviating agents, but instead towards the normative system as a whole.

Another aspect of change to take into account are the propagation effects of norm introduction and updates (see Ch. 2).

A possible approach to enable a runtime evolution of the normative system is to allow the designer to specify at compile-time a *normative transition function* that specifies the feasible norm changes and the conditions for their realisation [7, 74]. This approach however assumes it is possible to anticipate each normative update that the system may need.

7 Mapping to Other Perspectives

In this section, we discuss some approaches and conceptions of multi-agent systems that bear some similarity to regulated multi-agent systems (MAS), but turn out to be fundamentally different. In all of these, a regulated MAS-based modelling approach would potentially be a better fit.

7.1 Sociotechnical Systems

Traditionally, the field of sociotechnical systems has been concerned with the interplay between humans and technology in an organisational setting: how technology affects humans and vice versa. It developed with the recognition that the efficiency-focused design of work removed from the concerns of the workers or end-users (the social components) and the culture of the organisation tends to end up being self-defeating. This theme was picked up in software engineering in two major ways: (1) how to understand, elicit, and model the requirements for sociotechnical systems and how to manage change [53, 33, 2], and (2) how to model sociotechnical systems themselves [83]. Often, there was substantial overlap between the two, for example, as in i* [83]. Closer to the multi-agent systems literature, Flores et al.'s well-known work on the design of work [23] is best seen in the light of sociotechnical systems.

However, unlike regulated MAS, models of sociotechnical systems developed in software engineering are invariably conceptually centralised. This means that conceptually there would be a single thread of control in the system. Baxter and Sommerville [2], for instance, ascribe goals to the system; the system would potentially adapt in pursuit of its goals. Further, in their conception, systems are hierarchically decomposed into subsystems, each of which may either be a social component or a technical component. Yu's [83] models are peer to peer; however, his approach is mentalist and therefore implies conceptually centralised systems.

Models of sociotechnical system may well support physical distribution via the notion of components. One could then talk about the "interactions" among the components. These interactions, however, are merely *technical* in the sense that they are merely the means to an end—the system goals. Thus even though a system may be physically distributed, it remains conceptually centralised. In fact, it turns out that all of traditional software engineering has a conceptually centralised perspective on systems [14]. The reason is that even the most fundamental conception of systems in software engineering is *machine-oriented*. In other words, the primary objective of software engineering is to design machines that transform inputs to suitable outputs. The conceptual centralisation in software engineering is not surprising given that sociotechnical systems research, even outside of software engineering, has largely confined itself to organisational settings.

Regulated MAS represent a more general model for sociotechnical systems. With regulated MAS, one can model inter-organisational settings, which cannot be modelled with current approaches. Further, one can potentially argue that even for intra-organisational applications, regulated MAS-based models would be superior to the top-down models of traditional software engineering, because, after all, even *within* an organisation there would be multiple autonomous agents. Many of the approaches advocated by those in sociotechnical systems research and software engineering (for example, ethnomethodology and other methods from the social sciences) could potentially be employed toward the design of regulated MAS just the same as a centralised sociotechnical systems.

7.2 Agent-Oriented Software Engineering

Can agent-oriented software engineering (AOSE) be employed for designing regulated MAS? The answer is no. Although, many AOSE approaches give prominence to interaction, they take a conceptually centralised perspective. Some AOSE approaches are logically centralised approaches geared toward efficient problem-solving. Jennings [41], for example, describes a scheduling problem that is addressed by *distributing* it across intelligent agents. Both Zambonelli et al. [84] and Vázquez-Salceda et al. [77] acknowledge the distinction between open and closed systems and emphasise interactions. However they falter in important details, betraying if not conceptually centralised mindsets, at least considerable conceptual difficulties. For example, Zambonelli et al. (p. 328) identify the "resources that the MAS can exploit, control or consume when it is working toward the achievement of the organisational goal." Vàzquez-Salceda et al. model the objectives of social systems as goals; further, the social systems are themselves controllers (p. 338): "Facilitation roles are usually provided by agents controlled by the society, and follow a trivial contract." Tropos [4], another prominent AOSE methodology, is goal-oriented and advocates a top-down system design process starting with stakeholder goals and ending with the coded "agents".

7.3 Design Norms

A regulated MAS displays the following characteristics.

- The norms are social in that they are relations among agents.
- The norms are social in that the state of a norm would progress only due to explicit communication among agents. Logically, the social state of a regulated MAS is a conjunction over the states of individual norms. The specification of how communication affects the norms is referred to as a protocol.
- The social state of a MAS may be computed by observing the communications in the MAS. The social state is distinct from the internal state of any of the agents in the MAS; in general, there may be no overlap between the two (although in distributed systems, there would be no unique social state because of asynchrony; instead, there would be a local social state corresponding to the messages each agent observes [15]).

Commitments, for example, are created, discharged, delegated, and so on *only* by explicit communication—in distributed systems, via asynchronous messaging—among the agents. Commitment protocols specify how commitments among agents progress with interaction [82]. Analogously to commitments, Singh extends the treatment to other kinds of norms such authorisation, power, and so on [69].

The above characteristics set regulated MAS apart from approaches where norms are inserted into agent designs by *fiat*. This includes the social laws-based approach [63], which is essentially a distributed artificial intelligence approach. In the design-by-fiat approaches, the agents are not autonomous (they do not represent real world principals); they are instead agents in the technical sense. Social laws, which are sometimes referred to as norms, are essentially constraints on agent designs (specified with respect to a state space common to all agents). By contrast, norms in regulated MAS are not constraints on agent design; in fact, they make no reference to agent designs whatsoever.

Regulated MAS is a more general approach than social laws. One can potentially design agents to conform with the norms established during their interaction; however, one cannot apply the social laws approach to models systems of autonomous agents.

7.4 Compliance

Norms in regulated MAS are intimately tied with the idea of *compliance*. Broadly speaking, an agent is compliant with a norm if it does not violate it. Hence, compliance is fundamentally a runtime correctness criterion. Further, it can be determined by observing solely the communications of the agent (both to and from) [78, 70, 60] (this is because of the notion of social state discussed above). This is a crucial point: that compliance would be determined from observations implies that one does not have to look into agent designs to determine compliance (which would be impossible anyway in systems of autonomous agents). This should not be taken to mean that one cannot design agents for compliance. Given a set of norms, one can design agents to comply with the norms. Some refer to this design problem as that of compliance; we, however, reserve the term *compliance* for the runtime sense described above.

If an agent is noncompliant, a compensating norm may kick in, and if an agents violates that as well, then another compensating norm, and so on. At some point though, some norm for which no compensatory norm is specified may potentially be violated. At that point, we say that the violation must be handled outside the regulated MAS; in other words, it must be escalated to the surrounding organisational context [75]. Hence, it seems useful to frame a broader notion of compliance that would take into account the relations among norms (for example, via compensation) and its potential escalation outside the regulated MAS. Singh's idea of explicitly representing the context of commitments in a MAS as an agent [66] could be useful in capturing this broader notion of compliance. In [71], Singh et al. present several patterns of commitments that involve the use of the context agent.

Norms in regulated MAS serve as logical bases for compliance. They not only specify the conditions for compliance, but also who is responsible to whom. Compliance has received much attention in the business process community; however, this community mostly approaches it as a design problem, for example, as in [35, 64]. These approaches resemble the design-by-fiat approach discussed above: they lack the observational perspective, having no notion of communication and social state.

8 Demonstration: GOI a Sociotechnical System for Open Innovation

The *Green Open Innovation platform* (GOI) is a sociotechnical system to support "open innovation" [12] in the realm of sustainable economy. It enables business interactions among a community of firms and individuals who are interested in potential collaboration.[2]

GOI was originally conceived as some sort of social network portal with the standard "Facebook" functionalities—group definition, private mail, forums and "like" / "dislike" markings—on top of which there would be one interaction context for inscribing a "challenge" (either as an offer or as a request) and another interaction context for a "prediction market" (to show support to challenges).

In practice, though, the design evolved so that the platform is articulated around an "agreement space" that consists of four specialised contexts of interaction (agreement preparation, agreement negotiation, agreement monitoring, and quality assurance) that make use of multiple on-line services and repositories to support social coordination (not unlike what is advocated below in Section 10.2). Additionally, the platform is designed to support *APIs* for some standardised activities like crowd-sourcing and prediction markets, plus mash-ups and partnering for ancillary services like an employment market and some environmental certification services. Figure 3 sketches that setup. The system is designed to allow humans and software agents to participate.

The system may be recognised as a regulated MAS in as much as the platform provides the first three components of a normative architecture described in section 2 in P. 96. Namely, a virtual space, a common ontology and an interaction model. In fact, the platform provides an institutional infrastructure in the sense that it regiments ontology and means

[2] GOI is a proof of concept prototype for the development of a commercial sociotechnical system. It is an on-going project involving several private companies, NGO's and universities. It is partially funded by grants from the Spanish and regional governments.

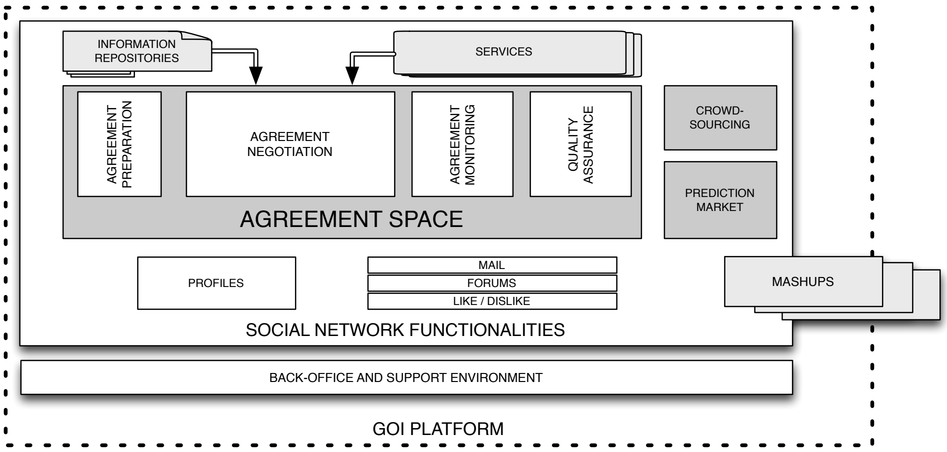

Figure 3 The GOI platform.

of communication, as well as the procedures that govern how to pass from one context of interaction to other and the procedures that certain processes must follow. For example, the platform regiments how challenges are issued, taken-up and monitored, or how to contest an active agreement. However, the agreement space also includes norms that may be violated as well as devices to contend with violations; thus containing the last two components of a normative architecture.

The agreement space, in broad terms, hosts a population of agents that are entitled to enter into agreements and are "active"—i.e., ready to be invited to an agreement, searching for partners for establishing an agreement, or being involved in the negotiation or in the enactment of an agreement—and are "ubiquitous"—i.e., agents may be involved in several agreements at a given time. The space is "open"—in the sense that agents may enter and leave at will—and it has some regimented ground rules on what are the primitive and atomic actions, the procedures that govern the basic agreement process cycle from start to end and the procedures to access and update GOI repositories and invoke services, as shown in Figure 4.

Perhaps the most interesting element of the GOI platform is that some agreements that are reached and executed inside the agreement space are in fact contracts whose clauses are negotiated among parties and their execution is monitored by the platform. The platform provides different means to facilitate this contracting. For instance, it has a repository of model contracts, whose clauses are standard and therefore negotiation is reduced to agreeing on parameters. Another resource is the availability of negotiation procedures that take the form of virtual "negotiation rooms" where a GOI member requests the platform a "room" to negotiate a contract. This member may want a room to hold an open call for tenders, hence may also request to have GOI staff run the tender for him (with software agents that perform the duties of, say, gate-keeper and auctioneer). Another member may, likewise, wish to negotiate one single model contract with N different potential partners simultaneously and would therefore require N rooms for one-to-one-negotiation, each with a software mediator with some explicit criteria for rejecting and admitting counteroffers, and a single arbiter to close the deal. Since not all contracts are honoured and some may be contested by other GOI members, the platform also provides means to resolve disputes, both in the form of automated ODR facilities or by making expert (human) mediators available. Finally, the platform keeps track of all actions where it is involved and therefore keeps information about participants

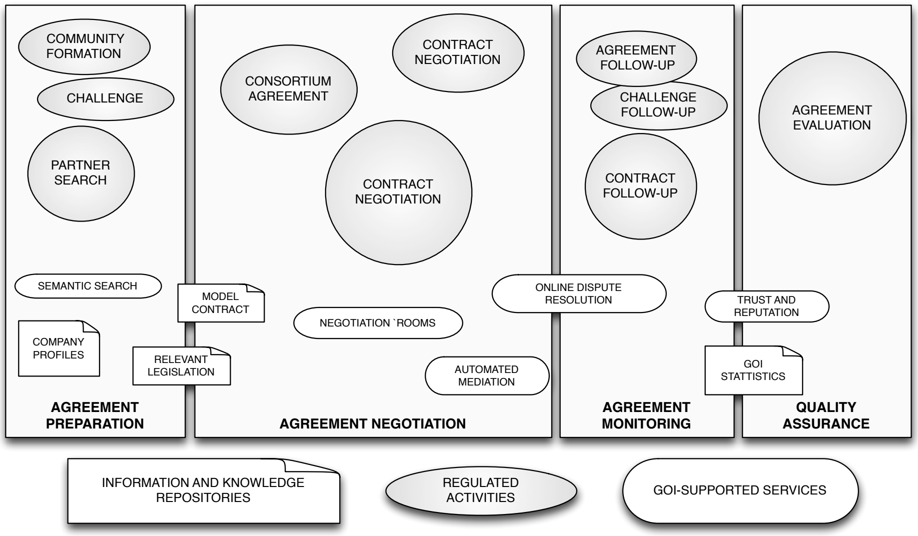

Figure 4 Main functionalities of the agreement space.

and transactions and uses it to provide different forms of quality assurance ranging from "blue book" rankings of members to specific trust and reputation measures that may be associated to particular agreements, types of agreements, communities and subcommunities, and so on. In other words, the GOI platform involves a regulated multi-agent system where each agreement includes its own normative content, is subject to some procedural and functional norms that regulate how it may be signed and how it should be executed, is also subject to the norms that govern malpractice and defaulting and, finally, some forms of quality assurance rules apply to it.

The GOI platform has a centralised design and its evolution is for the moment limited to the changes brought about by the addition of new services and interaction contexts. From an implementation perspective, the platform environment is also centralised, although agreements are distributed processes. Actually, there is also a central governance model for the basic operation, however each agreement spawns a (sub)context of interaction that is governed by its own norms and whose effects are, in principle, not propagated to the contexts of other agreements.[3]

9 A Map of Open Problems

This section outlines some open problems around the core notions of regulated MAS. The outline follows the three main phases of the regulated MAS lifecycle (design, enactment and evolution) and then focusses within those phases on some activities that are characteristic of normative environments in general. For each of those activities we mention those topics we find particularly relevant for norMAS and where we consider open problems abound.

[3] Each agreement generates a local institutional state that is part of the GOI institutional state (see Page 114) but the platform ought to guarantee the integrity and identity of agents and their institutional facts, thus some regimented control is imposed on the validation of agreements and their monitoring. Implementation follows the ideas discussed in [22, 19, 32]; hence, in essence, the agreement space is a large institution where each agreement is a new sub-institution that is specified and run peer-to-peer on demand.

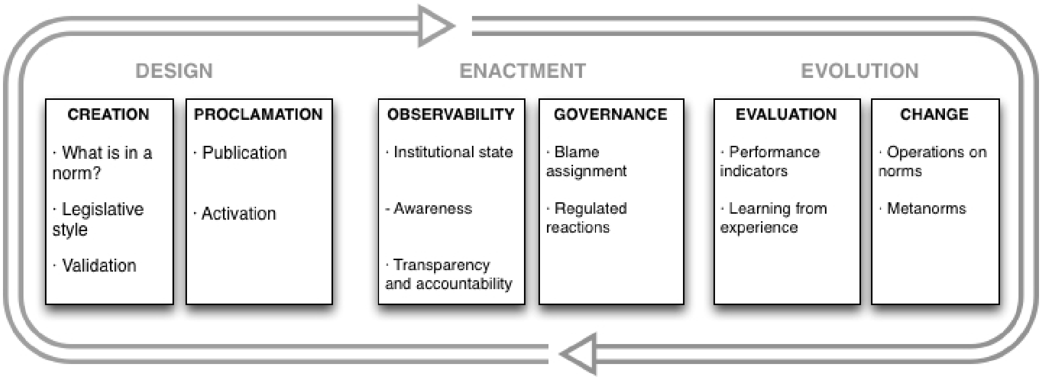

Figure 5 Some challenging topics within the lifecycle of regulated multi-agent systems.

Figure 5 summarises this outline. We elaborate on concerns introduced in Sections 4, 5 and 6 but note that while in this section we simply allude to some salient topics that we believe deserve a more thorough treatment, in the next section the authors of this chapter delineate some open problems that they find alluring.

9.1 Design Phase

Several design challenges emerge around the methodology for developing regulated MAS and the need for good enough metamodels to describe and specify them. In the case of methodology, the space for innovation is in dealing with those aspects that pertain to normative notions and how these are merged with the more conventional aspects for which methodologies have been proposed (see Section 10.2). In what corresponds to metamodels, the goal is to specify in a cohesive way all the components of the regulated MAS (see Section 5 and 10.1 below). The following subsections discuss several issues that need to be taken into account at design, in this section, however, we limit the discussion to design choices in two areas. On one hand, the *creation* of a boot-strapping nucleus of a regulated system including the particular norms and other regulatory devices (like normative roles, enforcement mechanisms or validation methods) that constitutes the original regulated MAS; and, on the other hand, a *proclamation* process that makes that original regulated MAS ready to be enacted and used by participants. Each of these areas include topics that are rich in concepts and complex in operationalisation, hence open for innovation.

9.1.1 Creation

Assuming proper methodology and metamodels are at hand, one still has to deal with the particular choices for each component of the regulated MAS in order to specify the system and make it operational. Let's review three areas of opportunity associated to the process of creation itself:

What is in a norm? The normative elements of a regulated MAS may have implicit requirements whose representation should properly take into account. These are a few that may be worth elucidating.

First, there are three key questions worth posing with respect to the intended *purpose* of the norms: (1) *What is the role that the norm is intended to play?* One may want norms to serve as a way of restricting unwanted actions and promoting desirable outcomes (hence appropriate descriptions of obligations and prohibitions are essential); likewise one could design them as a means to reducing the number of possible actions or outcomes in order to simplify the decision-making of participants (hence the representation of procedural norms needs to be paid special attention); moreover, norms may also be understood as a manner of creating a space for interaction (thus constitutive norms are paramount), or eventually the promotion of some collective objectives (in this case, then, such objectives should be properly captured in the form of the norms and their compliance incentives). (2) *What are the values that norms promote?* Values like fairness, trustworthiness, accountability are usually associated with the normative system as a whole but particular norms and their combination bias the system in one direction or another. Thus for example, full observability and strict governance may contribute to transparency, regimentation towards trustworthiness and unobtrusive governance towards flexibility. (3) *What are the pragmatic benefits of a norm?* The question here is to be able to establish a proper trade-off between the costs of observance and enforcement of norms and those of compliance and noncompliance. Some norms may have little effect on the reliability of the system but still impose agents an undue cost in deciding whether or not to comply with them; some norms may be practical for some population profiles and not for others and different sanctions may adapt better to some situations than others.

Next, there are three considerations with respect to the formal features of the *structure of the norm* to keep in mind to make sure that such structure is appropriate with respect to the desired expressiveness of the norm, the conditions for adoption of the norm and their compliance by agents and the governance mechanisms of the system: (1) The choice of the *syntax*, the *constitutive elements* involved (label, conditions of activation, deontic features, beneficiary, subject, ...) and the *ancillary elements* associated with the norm (linked norms, contrary to duty actions, ...). (2) The *crispness* of the statement of the norm and its applicability. In other words: Is the way that the norm is expressed and enacted precise enough for the fulfilment of its intended purpose? Is there an objective way of determining when an agent is complying or not with a norm? Does the expression of the norm achieve some desirable degree of flexibility (for discretional enforcement, to adapt properly to the evolution of population or changes of the application context)? (3) *Coherence of the normative corpus.* This has to do with the different formal properties that the system as a whole should exhibit. In some cases the preferred notion is that of logical consistency but in some cases it may suffice with a narrower notion of conflict-free sets of norms or, in more general terms, in choosing some particular notion of "consequence" that is appropriate for the system. Depending on those preferences, the designer would have to answer questions like: May conflicts be dealt with during runtime? Are there possibilities of deadlocks? Are there any norms that are impossible to comply with? How to enforce norms when the system has not proven to be conflict-free? Are norms conducive to the overall purpose of the regulated MAS? To what extent?

Finally, the design of the system should take into account *what an agent is intended to do with norms.* There are three capabilities that are assumed of agents in this respect: *Reasoning* about norms: What are the decision-making capabilities needed by agents in order to comply with the promulgated and active normative elements? To what extent are agents presumed to infer the state of the system in order to know whether a norm applies, has been violated and what the consequences of a violation might be? *Adoption* of norms: Is the ontology of the regulated MAS compatible with the norm (i.e., with respect to the objects, actions, roles, and such involved in the norm)? Is the governance structure appropriate for enforcing the norm (detection, blame assignment, sanctioning and reparation)? Is pertinent information about entitlements, co-dependence of normative elements, effects of noncompliance, and so on properly represented and communicated to implicated parties? *Compliance* with norms: What behaviour is perceived and by whom? What information about a norm should an agent be informed of: social values associated, whether or not it is active, conditions of application, regulated reactions, subjects of the norm, consequences of not-compliance, liability, enforceability conditions?

Legislative style. The designer needs to make choices about the normative features that have to be functional when the sociotechnical system is originally enacted and how those features should evolve once the system becomes active. Three matters of concern may affect those choices and all three have plenty of open problems.

Locus of control: The issue is to determine who controls the changes that take place in the normative system. One extreme is a "demiurgic" style: the designer that creates the regulated system has control of it as a whole and introduces changes as needed. The other extreme is where a minimal set of regulations are instituted and participants are able to introduce new norms and appropriate forms of observability and enforcement once the system is in operation. Because the reasons for introducing a change and the choice of the type of change that best applies, are manifold, the usual solution is an intermediate position. However, as discussed below in 9.3, finding out how to determine what those intermediate positions are, and how to implement them is almost unexplored territory.

Other design choices determine the *balance between regimentation and the different degrees of enforceability*. What are the matters that should be taken into account? For instance, considerations about accountability, robustness, and transparency may tilt the balance towards regimentation, while on the other hand addressing or allocating risk through constitutive conventions (e.g., requiring bonds and guarantees from principals, or relying on a conventional legal system to deal with severe transgressions), or the need for flexibility together with the presumption of reliable autonomous agents tilt the balance towards self-governance.

A third closely related matter is the *regulation attitude*. In this case, again, in most cases the final choice is bounded by two opposite styles. On one side is what we may call "preemptive enforcement" where—as in a typical Napoleonic legal tradition—presuming any agent would break the law if given a chance, the system designer anticipates all possible infractions, makes these and their ad-hoc corrective reactions explicit, and commits to a strict enforcement of this list. On the other side there is a "laissez faire" attitude where it s presumed that most individuals abide by the law most of the time. In this case—mimicking a Common Law tradition—undesired behaviour is expressed in general terms and only when someone is caught cheating and proven guilty after a due process, a harsh exemplary punishment is applied.

Validation of the design. With the remarks in Section 3 in mind, there is room for innovation with respect to testing and validation of the system from the formal as well as the engineering perspectives. For instance, the *off-line validation* of the sets of norms under different criteria and techniques, from model-checking to coherence theory. *Computational complexity* of norm-abidance and *scalability* of the system with respect to the increase of norms and agents. *Expressiveness* of metamodels with respect to the intended performance of the regulated MAS. *Reliability* of the operationalisation processes. Although we are referring here to validation at design-time, there are limitations to what may be tested and proven off-line, hence some *runtime validation* may be needed, should be considered as part of the design and then implemented. In these matters the key is reaching an appropriate balance between what may be proved and what is satisfactory from an engineering perspective.

9.1.2 Proclamation

This process involves two aspects: on the one hand, how participating agents need to be informed about the system so that they will be able to play accordingly, and on the other hand, what needs to be operational so that the system may be enacted. Mirroring standard legal practice we may distinguish two sources of design choices where interesting problems arise: publication of the conventions and their activation.

Publication. The designer will have to commit towards those elements that need to be working from the start of and will also need to decide for each one of these elements how to make them known, when, and to whom. A mere enumeration of the elements involved in publication is enough to show the richness of this topic: constitutive conventions (ontology, primitive and nonprimitive actions, interaction model, access requirements, entitlements), architecture of the system (governance model, dynamics), normative content (different types of norms and metanorms) and eventually, the operational semantics of the system. The "how" part has two dimensions: the degree of formality (logical or otherwise) of those components and the process by which the components are made part of the system. Another aspect where innovation is needed is in the ergonomic side of communicating those elements: what type of expression, syntax, interface are appropriate, when, and for whom.

Activation. Some conventions—including most regimented constraints—are established at design time to become active the moment the system is enacted and hence are applicable from the start to any agent that intends to participate. However, while the system is being enacted, norms may be added, modified or revoked and these situations start to apply to participants at some point. Generally speaking, such activation may be triggered either by time (e.g., so many days after it is published) or by an event (e.g., once a commitment is made). The challenges in this topic come from different sources: from the regulated MAS perspective, the immediate ones are how to validate that an agent that intends to participate complies with the conventions that regiment its admission, and then how the system deals with latent norms. From an individual agent's perspective, how is an agent made aware of those norms that may be applicable to it.

9.2 Enactment Phase

We do not touch upon the computational and implementation aspects of enactment, only on some topics that may be interesting from a regulatory perspective while the system is active. Furthermore, for sake of brevity, most of our comments refer to noncompliance with norms (and the corresponding negative sanctions), although they apply *mutatis mutandi* to the compliance of actions that have a positive reward associated.

9.2.1 Observability

Institutional State. The essential feature of regulated systems is that there are norms that individuals may or may not infringe. Hence, at some points, although individuals decide whether to comply with *applicable* norms, the system as a whole, or its enforcement devices—and usually some participant agents as well—may have to assess that a compliance or noncompliance took place. Such assessments involve the difficult technical (operational) problem of representing and updating the "institutional state" of the regulated MAS (see Section 7.3). In other words, what are the values of those variables that represent the institutional facts at a given moment, how atomic actions are filtered into the regulated MAS and how those actions that are deemed "institutional" modify the value of those variables.

Assuming that the institutional state is properly represented and implemented, the designer still has to choose how this state is accessed by participants while actions are taking place. In other words, the regulated MAS needs to address, the *ex-ante* aspects of "awareness"; and the two complementary *ex-post* aspects of "transparency" and "accountability". These three types of aspects are solved by regulating what part of the institutional state is revealed, to whom, when and how.

Awareness. The challenges in the *ex-ante* phase of compliance assessment reside in what is revealed before an action takes place.

From an individual agent's perspective, that revealed information is needed to support two decision-making tasks: On the one hand, the individual agent needs to be aware of that state of the system to realise what active norms apply to it in order to decide whether or not to perform an action that may infringe those norms. On the other hand, an individual needs to be aware of the state of the system in order to form expectations about what may happen then: who may act, what actions may be attempted and what effects these may have.

From the system's perspective, the challenge is again twofold: to determine what actions are feasible (nonregimented), and to determine which agents are subject to active (nonregimented) norms and therefore have the possibility of complying or not with it.

Transparency and accountability. The first type of *ex-post* aspects, "transparency" refers to what is revealed (to individuals and to the system) about actions that take place and their institutional effects. "Accountability", in turn, refers to information about who performs an action and who is affected by that action. The two of them together become input for determining awareness in subsequent institutional states. Both of them are, evidently, key for enforcement and should consequently be in line with the enforcement model of the system and the enforcement style we mentioned above. Both are particularly challenging when the regulated mulitagent system involves nested or concurrent regulated MAS where commitments established in one regulated MAS may have effects in other regulated MAS. Transparency and accountability also need to address the complementary aims of "need to know" and "need to share".

The opacity of some actions may be appropriate. For instance, in some mediated negotiations, only the mediator is informed of offers and counteroffers and each party is unaware of what the other part is actually proposing. In some cases, while opacity is required for some purpose, it needs to be compensated by some means. For instance, even if a noncompliant action is itself opaque to law enforcer agents, an infraction may be inferred by these agents if they perceive some effects of that action, or they are informed of the infraction by other agents that witnessed or inferred it on their own; likewise, even if a punishable action is perceived by an enforcer agent, the agent may decide—or be compelled—to ignore it. Opacity may also have adverse effects. For instance, if an infraction is not immediately dealt with, its indirect consequences may be difficult to foresee and contend with.

9.2.2 Governance

This area includes those actions that follow the assessment of compliance. Namely the processes of determining whether or not an infraction took place—analogously, a reward-deserving action—and then react accordingly. In the judiciary tradition, governance involves three main processes: prosecution, trial, and punishment. Although some of these issues have already received attention from different perspectives, the topics are rich and still largely open for regulated multi-agent systems. We will next describe the most salient ones in two areas, blame assignment (involving prosecution and trial to some extent) and reactions (including punishment and other). As with the previous paragraphs, our comments are biased towards noncompliance and sanctions but similar ones would apply to rewards and desired actions.

Blame assignment. Given that a punishable (or reward deserving) action has occurred, the challenges reside in determining that the action took place and should be punished (or rewarded), determining who is involved and who is responsible for the infraction (or reward). The assessment of infractions and culprits will depend on the observability of actions and ultimately on the enforcement model of the system and the enforcement style. For instance, if the enforcement of some norms is delegated to enforcer agents, when these agents observe or infer the infraction, they would need to be able to identify the beneficiaries along with the casualties associated with the infraction. Provided a good level of accountability is available, these enforcers should then be able to assign blame. Once the infraction is acknowledged and culprits identified, reparatory actions and sanctions are enacted according to the norms that regulate these processes. The process of determining who the culprits are is not necessarily straightforward because, depending on the observability or the infraction, the identity of perpetrators may not be revealed, and even if the agents who perform the invalid action are properly identified, it may still be necessary to prove who the actual guilty parts are. In this case some sort of *due process* needs to be activated. When a regulated multi-agent system contemplates the existence of a due process, several components need to be in place. Namely, some notion of proof that an infraction has occurred, the resources to elaborate and validate that proof and the procedures that correspond to (i) bringing charges and evidence against suspects, (ii) defending innocence against charges (iii) evaluating evidence and applicable norms and (iv) formulating a resolution about the innocence or guilt of the accused.

Regulated reactions. The types of reactions that are worth studying may be organised in two large blocks: punishment and reward on one hand and damage control.

Punishment and rewards. How the punishment is expressed (threat, argument ad baculum, and others), Grounds for punishment (what needs to be proved to deserve that punishment or reward). Purpose of punishment (to teach, to encourage, to retaliate), types of punishment (direct or indirect, private or public, with rhetorical information associated, with social or monetary cost). Management of punishment (whether it is applicable on the spot, delegated to another regulated MAS, and so on).

Damage control. Identify direct and indirect effects of an infraction. Measure costs of transgression, evaluate costs of detection, blame assignment, and punishment. Identify and implement reparatory actions (fix damages, compensate victims). Here belongs the challenging world of *contrary-to-duty obligations* that has received ample formal treatment and whose implementation is nontrivial (see [13, 9, 34]).

9.3 Evolution Phase

Several remarks on this phase were made above in Section 6 and more will be made in the last section of this chapter. Here we simply mention a few more open topics. and for the sake of presentation we mention topics about only two aspects: performance evaluation and change.

Note however, that from the design perspective (even when evolution is postulated as a bottom-up process) it is advantageous to identify the several conditions that might make the regulated system evolve and, consequently, include devices to handle that evolution. One may argue that the type and protagonism of those devices depend largely on the expected evolution of the system, and different types of devices will need proper grounds to determine their application. In general, one would need to identify (in the metamodel and the methodology) (i) a reasonable list of devices, (ii) for each device, the type of situations that justify its use, (iii) the elements of the system that are involved in those situations and (iv) the interplay of those elements in a given situation.

As an example to illustrate the richness of the problem, note that one of the many devices to make a regulated system evolve is to change some of its norms; moreover, one of the ways that norms may be changed is directly by the system (not by participants), and those changes might be advisable, for instance, because performance of the system has decreased due to, for example, changes in the population profile or the environmental conditions. Now realise that, even in this rather simple case where we assume that *one* evolution device of the system is to change norms, in order to decide what norms to change we would then need to foresee, at design time, that performance of the system can be measured, that changes in population and environmental conditions may be assessed, that the relationship of those changes with performance are made explicit, that the set of norms that may be modified (added, deleted, changed) in order to achieve the desired performance levels can be determined and, finally, that the actual modifications may be accomplished.

9.3.1 Evaluation

Performance indicators. It is not unusual to have sociotechnical systems involving stakeholders with competitive goals and thus having regulations intended to achieve equilibria of different sorts. However it is not always clear how these equilibria might be identified without an explicit reference to some variables involved in the operation of the system and their combination as indicators that are meaningful in terms of the objectives of the stakeholders or the system as a whole. In general, variables and indicators are useful to assess the quality of the system (or parts of it) and to choose between alternative implementations of regulations at design time and to guide evolution at runtime. It is a challenging task to deal with performance indicators that evaluate how a system performs or improves in important but elusive qualities such as fairness, trustworthiness, and accountability.

Learning from experience. As suggested above, the choice and combination of performance indicators depends not only on the objectives but also on the devices that are available to make the system evolve and on the elements that are involved in the application of such devices: from changing some parameters of a norm, to changing the set of norms. Part of the tuning—not only of the original normative system but notably also of the evolution mechanisms themselves—may be achieved at design-time by stress-testing and running simulations; and there is ample opportunity for development of the current practices and tools.

9.3.2 Change

Operations on norms. A full typology of operations on the set of norms (promulgation, amendment, suspension, abrogation, annulment, ...), as well as the operations on their governance aspects and their implementation is still to be attempted from the regulated MAS perspective. Some works in this community suggest [5, 74, 36] that a systematic treatment of the different operations is far from trivial.

Metanorms. Another line of work that is largely open for research is to include all those aspects that determine the dynamics of the system as a distinct core component of a regulated MAS as postulated in Section 6 (p. 105). Several approaches may be taken and a few have already deserved attention like, for example, the use of metanorms to choose among predefined sets of norms [1, 7], or the use of case-based reasoning to introduce new norms when conflict among norms are detected [52] but a holistic proposal towards a general normative transition function, as suggested before in 10.4.1, is still to appear.

10 Authors' Perspectives

This section is meant to complement the rest of the chapter by including personal views of the authors. Each author has chosen topics, length and structure.

10.1 Noriega: In Light of Applications

I would like to make the following remarks under the light of regulated multi-agent systems that are going to be used in open sociotechnical systems that are intended to work in the real world.

In this context, what I believe to be the most fundamental task is to build a general framework on top of which actual normative multi-agent systems may be specified and implemented. By a framework I mean three main components: a normative architecture or "metamodel" for the specification of normative multi-agent systems, the computational counterpart of this metamodel that would allow to implement and run such specifications, and the methodologies to guide the actual implementation of such norMAS.

Following the experiences reported in [19], I believe that such a metamodel can be built along the lines described above in sections 3, 4 and 5, in order to support the five components enumerated in Section 2. The computational counterpart should produce an "institutional environment" where particular regulated MAS are enacted, all regimented constraints are enforced and several regulated MAS may concurrently exist. That implementation may opt for a centralised governing of the full environment (including the possibly several regulated systems embedded in the shared institutional environment) or choose to have a distributed architecture to handle particular norMAS [59]. The methodologies should facilitate four activities: the proper specification, implementation and testing of particular norMAS—including whatever agents are needed to perform those regulatory roles entailed by the specification—and, in the fourth place, assess the adequacy of the implementation of the norMAS with respect to the intended functionality in the world.

If taken to heart, this task involves several challenges that being already in this community's agenda, as suggested in the previous section, are far from being won. The ones I see as the most significant are the following:

10.1.1 Achieving Good-Enough Expressiveness and Automation of the Normative Languages

Let's assume that the framework includes a rich enough meta-modelling ontology to capture several interaction models, normative corpora and governance models. Then, for the specification of particular regulated MAS, it will be necessary to use several normative languages (and many "types of norms", e.g., Crawford and Ostrom [17]) and possibly non standard notions of "consequence", that should be compatible with the features discussed in 7.3 and 7.4. But, in addition, those norms will need to be accompanied with other linguistic and formal construct so that the *declarative* and *inferential* features of such norms carry the intended pragmatic load within the normative MAS. So, for instance, in order to deal with *procedural norms*, it may be advantageous to use, say, commitment-based protocols, hence one has to introduce a proper language [27, 82] as part of the regimented constraints. Likewise, to implement *contrary to duty functionalities*, one needs declarative languages that accommodate the subtleties of features like the set of linked norms, including sanctions and reparations, as well as some automated means to support the complex inferential processes involved in blame assignment and proper reparation.

The implementation of these features should take into account, not only what the regulated systems themselves should be able to accomplish at run time, but also how agents (software as well as humans) become informed of all those features; in order to take them into account both at design and at running time.

10.1.2 Designing for Noncompliance

The framework, as I see it, serves a general-purpose boot-strapping function, in the sense that it should support particular normative MAS and provide the environment where actual normative MAS are embedded. Therefore, the metamodel should support the modelling of different governance models and provide the *expressive features* to specify those different governance models, and the framework should support their *automation*. Moreover, the framework should support the interoperation of several of these regulated MAS in a common environment.

In these conditions, regimented governance is unavoidable for certain aspects of the environment (e.g., common ontology, atomic interactions, social semantics, how new regulated MAS are embedded in the environment) and maybe also within the particular regulated MAS. However, in addition, in most regulated MAS, one would need to have nonregimented conventions that deal with the forms of enforcement described in page 104 and in Section 4.

In practical terms it is unlikely that there may be a general treatment of governance modelling and implementation. Nevertheless, it is still hard but worth attempting to develop some sort of "standard" devices to deal with the several aspects of governance (detecting, ascertaining, evaluating, assigning blame, applying sanctions) and assemble such devices as the enforcement model for a particular regulated MAS. Assuming regimentation is properly handled in the framework (like suggested in [19]), there are two ways to approach the enforcement of nonregimented regulation. The first one is to rely on an omniscient filtering device—like the one needed for regimentation—that automatically produces all consequences of a new fact, thus updating the institutional state [74, 40, 31]. The second one is to rely on some form of law enforcement roles that are capable of perceiving some actions and react with or without discretion in accordance with the relevant regulations; this applies also to informal sanctions, applied by individuals who may not be playing proper law-enforcement roles at all.

Another path worth exploring is that given that transgressions will take place, the governance model may in some instances grant the offender the benefit of doubt and allow the wrongdoer to argue the case. This would serve two purposes, allow for some useful forms of ambiguity and to improve regulations.

10.1.3 Designing with Ambiguity

Conventional legal institutions incorporate ambiguity in ordinary legislative practice for two main reasons: to prevent overregulation and to allow practitioners and enforcers some leeway for interpretation. These reasons also suggest the need and advantages of including ambiguity as a design assumption for regulated MAS.

In general terms regulated MAS come across “opaque” contexts where some undesirable situations are difficult to properly identify at design time. Some obvious situations are: (i) Regulating the wrong problem. For example in the *mWater* system described in the next chapter, it is easy to focus only on pricing conventions when the real problems reside in the lack of supply and in the conflicts provoked by the use of traded rights. (ii) Regulating the wrong population; for instance, unwittingly ruling out human agents, by setting the bidding-clock pace in an auction too fast. (iii) Changing population: In the open innovation scenario of Section 8, it is rather likely that the client base becomes more international and more sophisticated as the level of activity increases. But then standard contracts and quality assurance criteria would have to be tuned to the new situation [7]. (iv) Unforeseen undesirable outcomes; In automated trading, software trading agents may have equivalent bidding algorithms and enter into unending ties. (v) Volatile contexts where the grounds for choosing particular actions may change depending on circumstances that, because of their unpredictability or variety, are not worth hardwiring in stable norms, but rather be localised to those contexts.

Mirroring the points just made, the following are obvious heuristics to start bringing ambiguity into design: identifying volatile contexts in key business processes; building malleability in constitutive norms (for instance, committing to a good-enough ontology while including norms to update it); separating stable norms (that may be hardwired as regimented or fully specified enforced regulations) from norms that are better represented in the system as norms that govern decision-making of law enforcers; using flexible-enough governance models and choosing performance indicators that measure different types of “fitness” of the regulations.

At any rate, whatever heuristics are devised, the components where ambiguity will need to be instrumented are the norms that govern the evolution of the system. And there is where ultimate design with ambiguity will be achieved.

10.2 Chopra: Social Computing, A Software Engineering Perspective

The nature of applications is changing. Earlier they were logically-centralised; now they are becoming increasingly interaction-oriented. Social networks, social cloud, healthcare information systems, virtual organisations, and so on are evidence of the shift. In such applications, *autonomous* social actors (could be individuals or organisations) interact in order to exchange services and information. I refer to applications involving multiple autonomous actors as *social applications.*

Unfortunately, software engineering hasn’t kept up with social applications. It remains rooted in a logically centralised perspective of systems dating back to its earliest days and continues to emphasise low-level control and data flow abstractions. In requirements engineering, for instance, the idea that specifications are of *machines*, that is, controllers, is firmly entrenched. Software architecture applies at the level of the internal decomposition of a machine into message-passing components. In other words, it helps us *realise* a machine as a physically distributed system. However, the machine-oriented worldview cannot account for social applications in a natural manner.

I understand *social computing* as the joint computation by multiple autonomous actors. By "joint", I refer simply to their interactions and the *social relationships* that come about from the interaction, not necessarily cooperation or any other form of logical centralisation. In fact, each actor will maintain its own local view of the social relationships—there is no centralised computer or knowledge base. The relationships themselves may take the form of commitments, trust, or some other suitable social norm. The purpose of the computation may be to loan a bicycle or a couch to a peer, to schedule a meeting or a party, to carry out a multiparty business transaction, to provide healthcare services, to schedule traffic in smart cities, to manage the distribution of electricity in smart grids, to build consensus on an issue via argumentation, or globally distributed software development itself—*anything* that would involve interaction among autonomous actors.

Clearly, we are already building social applications, even with current software engineering approaches. For example, online banking is a social application in which a customer interacts with one or more banks to carry out payments, deposits, and transfers. Social networks such as Facebook and LinkedIn facilitate interactions among their users. However, just because we can build social applications, it does not mean we are building them the right way. Right now, all these applications are built in a heavily centralised manner: banks provide all the computational infrastructure; so does Facebook. Users of these infrastructures are just that—*users*, no different from those of an elevator or an operating system. In other words, current software engineering produces only low-level technical solutions.

My vision of social computing instead embraces the social. It recognises the autonomy of actors. Instead of control flow or message flow, it talks about the meanings of messages in terms of social relationships. Computation refers to the progression of social relationships as actors exchange messages, not to any actor's internal computations (although these too could be accounted for). The different aspects of my vision constitute a challenging research program. What form would specifications of social applications take? What would be the principles, abstractions, and methodologies for specifying social applications? On what basis would we say that an actor is behaving correctly in a social application? How would we help an actor reason about specifications of social applications with respect to its own goals and internal information systems? What kind of infrastructure would we need to run social applications? The answers to these questions and the realisation of my vision will lead to a software engineering vastly more suited to social applications.

10.3 Fornara: Formalising and Executing Open Normative Systems

The process of formalising at design time and executing at run time open interaction systems where the behaviour of agents is regulated by norms, presents interesting open issues.

A first one regards how to choose and use existing formal and programming languages and architectural solutions for defining and implementing in efficient and effective way open interaction systems. Those type of systems are composed by many concepts that evolve in time on the basis of the events that happen and the actions performed by the agents, therefore one problem is efficiently represents the evolution in time of their state. A crucial requirement in the specification of norms is that their content should be a description of the action that should, should not, or can be performed by the agents: that is a description of a *template* of the action. In order to let the autonomous agents to exploit their capability to dynamically plan their action by taking into account the norms that constrain their behaviour, it would be better that those template should not describe the actions in all details. This by making it possible that different concrete actions will match with the described template. For example a norm may express the obligation to pay a certain amount of money to Robert without specifying the actor of the action and making it possible to perform the payment in many different ways, like for example by bank transfer or in cash or by cheque.

In those type of systems it is necessary to implement specific components for realising the following required functionalities. First of all in order to *enforce* those norms it is necessary to be able to efficiently *compare and match* the real actions that are performed by the agents at runtime with their description formalised in the template of the action that is used as content of the norm. Practically it is also necessary to define a component able to *monitor* and to *regulate* (in the case of power) the evolution of the state of the system.

At least two more languages turn out to be necessary, one for the formalisation of the concepts the other for the implementation of the functionalities. In this case fundamental problems are how to combine them, and how to decide what to represent in a given language and what in another.

Reusability is a strong requirement in the development of software systems, therefore the definition of design mechanisms that make it possible to *re-use* at least part of a given specification of a system is another important challenge. The process of designing and developing open systems has to take into account also the design and development of the *agents* able to interact through those systems. The openness of those systems and the fact that they are running in an open network like Internet, implies that one agent may decide to interact simultaneously with different systems and that one system can be modularised in different components and distributed on different platforms, therefore a critical issue is the possibility to *combine* different specifications with few or none added re-design or re-programming work.

An important issue in open systems is related to how agents can perceive the shared state of the system and in some cases, when declarative communicative acts have to be performed (for example declaring an auction open) to use them for interacting with other agents. Regarding this aspect there are interesting existing approaches and existing frameworks coming from studies on distributed event-based systems and *environments* [81]. Agent environments can be considered an interesting architectural component that can be used for *mediating* agents action and for enabling agents to perceive the state of the interaction. The functionalities of the environment can also be extended to realise the *monitoring* of agent actions and the realisation of concrete mechanisms for *norm enforcement*. In this context, an interesting challenge would be the study of how to integrate and extend existing formal models and frameworks for the realisation of agent environments [6, 58] with the concepts and functionalities required by open normative systems.

One possible approach for trying to tackle some of the previously described challenges consists in formalising norms, and the corresponding obligations, prohibitions, permissions, and institutional powers, using *Semantic Web* technologies. In particular by using OWL 2 DL, a *description logic* language recommended by W3C for Semantic Web applications, for the specification of the main concepts of the model.

The main advantage of the choice of these languages is first the possibility to re-use ontologies and combine them thanks to the fact that two or more ontologies can be simply merged by taking the union of their axioms [39]. Another advantage is the possibility of using the semantics of the concepts used to describe the template of the actions contained in the content of the norms to effectively match them with real actions that happen in the system; for example being able to deduce that a bank transfer or a cash payment are both payment. From the point of view of the technologies and tools available for supporting the use of Semantic Web Technologies, important advantages are: (i) the availability of well studied and optimised reasoners (like Fact++, Pellet, Racer Pro, HermiT) for deducing knowledge or for checking the consistency of certain set of norms; (ii) the possibility to use tools for ontology editing (like Protégé) and library for automatic ontology management (like OWL-API or JENA).

Given that OWL 2 DL is mainly a language for expressing knowledge bases it may not be expressive enough for implementing running dynamic interaction systems, therefore a further crucial challenge is to study how to effectively and efficiently combine OWL 2 DL ontologies with *rule languages*—like Datalog, SWRL (Semantic Web Rule Language), or RIF (Rule Interchange Format) that is a W3C Recommendation from 22 June 2010—and with programming languages—like Java—for representing the dynamic evolution of the state of the system and for monitoring and enforcing the norms. Some preliminary studies on how to use Semantic Web Languages for realising open systems are presented in [28, 26].

The approach of defining different artificial institutions and then to combine them to realise different concrete open system is one of the possible approach that tries to tackle the problem of re-usability of artificial institutions. An open and crucial challenge is to study the process of transforming a conceptual model of an artificial institutions in a concrete running software where the interaction among autonomous agents can actually take place.

Concerning the challenge of situating open systems in agent environments a possible approach could be the introduction of a formal description of spaces of interaction and objects as first-class entities that shape the environment and describe all its structural components, including norms. A crucial challenge in this approach is managing AI and spaces interdependencies in terms of events and norms that regulate those events. Initial studies in this direction are presented in [73, 29].

10.4 Lopes Cardoso: Achieving Open Normative Environments

Most approaches to modelling normative multi-agent systems are based on statically defined normative scenarios. Even when assuming an open stance in terms of the nature of interacting agents (which are seen as heterogeneous self-interested entities that cannot be assumed to be benevolent), such approaches do not accommodate a normative-level autonomy [79] of interacting agents. As such, when entering the system agents adhere to the normative scenario that has been predefined. This is the approach we see in organization-oriented models (such as [18]) or in dialogical electronic institutions [54].

Looking at *open* normative environments from an extended perspective, norm-autonomous agents should be able to choose the norms they wish to govern their relationships. Therefore, it is not only the origin or benevolence of agents that is not certain (which is a must in open multi-agent systems), but more importantly the norms that are to be handled by the environment cannot be totally predicted in advance. It is thus important to think of the normative environment as providing infrastructures for fulfilling two distinct and complementary roles:

- *Normative setup.* How can agents be assisted in their effort to specify the norms that better fit their intended interaction scenario?
- *Norm monitoring and enforcement.* What kinds of mechanisms can the environment employ with the aim of monitoring and enforcing norm-regulated relationships?

While the latter has been widely addressed by several researchers working both on monitoring (e.g., [51, 48]) and enforcement (e.g., [57, 24, 37, 49]), the issue of runtime normative setup has been mostly neglected—as mentioned before, designed normative environments are typically closed as far as the normative space is concerned. Nevertheless, this issue is important in application domains configuring regulated coalition settings, such as agent-based electronic contracting or Virtual Enterprise formation.

10.4.1 Environment Design and Normative Evolution

The study of the *environment* as a first-class entity in multi-agent systems (MAS) is quite recent [81]. A role that the environment may fulfil is that of normative state maintenance, by employing appropriate norm monitoring policies. This has been identified as a viable alternative to deploying "institutional agents" responsible for this task: the governing responsibility is transferred to the environment itself [61]. Typically, a rule-based infrastructure that defines reactions to events is the approach taken to realise this kind of environment, allowing not only to regulate agent interaction but any action that is taken within the environment.

Along this line, *coordination artefacts* [56] have been proposed as abstractions encapsulating and providing a coordination service that agents can exploit in a social context. Coordination artifacts aim at enabling the creation/composition of social activities and at ruling those activities, from a normative perspective. Several proposals have been made to conceive artefacts of several types. This includes the use of organizational mechanisms to address the normative aspects of a MAS [40], mostly from a normative state monitoring perspective.

The structuring of a normative environment into different social contexts has been targeted by a number of researchers. *Institutional spaces* [72] are an approach to segment the environment into different institutions; these benefit from a common environment infrastructure in terms of perception and evolution models. Institutions can also be empowered to govern other institutions [16]. A slightly different perspective is to consider an institutional environment as being composed of several *normative contexts* [47] governing different but sometimes interdependent social relationships. A hierarchical organisation of contexts (or institutions, for that matter) allows designing the environment with norm inheritance models in place.

A few researchers have tackled the problem of changing the norms of the environment at runtime. Theoretical approaches assign to *constitutive norms* the role of defining the possible changes to the normative system [3]. More practically-oriented efforts consider defining appropriate constructs that allow the system designer to define at compile time a *normative transition function* that specifies the possible norm changes and the conditions for their realisation [7, 74]. Letting the agents choose at runtime the changes to introduce in a normative system (defined in terms of a dynamic protocol) specified at design-time has also been suggested in [1]. Again, the possible options are application-specific: a set of possible values for specific fluents are the *degrees of freedom* agents have when proposing protocol modifications.

10.4.2 Challenges and Perspectives

The notion of an *open normative environment* tries to look at the environment from outside any preexisting organisation, putting the emphasis instead on infrastructural components that allow *normative interactions* to take place. By normative interactions we mean not only interactions governed by norms, but also norms that are established by a deliberative process based on interaction.

Given the current approaches to address normative environments (whether disguised as artificial/electronic institutions or referred to as environments in their own right), the openness of multi-agent systems is not fully addressed. We argue in favor of the need to develop appropriate infrastructures for enabling runtime creation of normative specifications. Although, as mentioned above, there are some approaches in the literature that deal in some limited way with dynamic adaptation of norms, this issue has not yet been addressed from a domain-independent perspective.

Having software agents that are able to deliberatively establish on their own some normative specification of a norm-regulated relationship is a challenging task. Although there are several relevant research contributions regarding normative reasoning (mostly from a norm compliance perspective), the specification of appropriate infrastructural components facilitating normative setup is lacking.

Approaches such as coordination artifacts have been exploited as mediated interaction mechanisms. A similar mechanism may be explored for assisting the setup of normative relationships. The use of normative frameworks is another promising approach to ease this task. Norm inheritance or adaptation mechanisms can be explored. To this end, insights from legal theory are certainly pertinent sources of inspiration to develop environments that include such facilities.

10.5 Singh: A Normative Basis for Trust

Open settings, where norms apply, inevitable bring out the problem of decision-making: How can each party decide on how it should engage the others? Trust is a key ingredient in such decision making, leading us to another question: How can each party determine how much trust to place in another autonomous party? To be an effective basis for decision making, the estimation of trust must incorporate (1) the interaction—task or transaction—being considered by the decision maker, (2) the social or organisational relationships, and (3) the relevant context.

The following are the main approaches to trust today.

Today's distributed computing approaches hard-code some patently naïve assumptions, e.g., about the effectiveness of certificate chains [8]. Thus, they provide little or no rational basis for decision making in realistic settings.

Today's analytics-based approaches seek to estimate the trustworthiness of a party based on an analysis of its attributes, behaviour, and relationships [30, 44, 80]. However, existing approaches largely hide the deep structure that one would informally associate with trust in natural human and organisational settings. In particular, these approaches provide simple calculi that associate numeric measures or qualitative descriptions of trust that seek to summarise the trustworthiness of a party as a single real number or nominal value.

Today's cognitive approaches build ontologies and knowledge models of contexts and situations but in a way that gets into the particulars of a domain of application. Further, they provide complex definitions seeking to formalise how humans subjectively understand trust, but such definitions call upon concepts for which we cannot easily obtain any data to use as a basis for prediction or analysis [11, 10]. Thus, although the cognitive themes and domain models are useful, such approaches are not easy to apply computationally.

I propose a research direction that seeks to address the above gap in the modelling of trust in a manner that exploits the inherently normative nature of multi-agent systems to address how we can analyse and engender trust. If our collective research efforts succeed, our main contribution will be a principled approach to trust that provides rich abstractions for modelling tasks and social or organisational relationships that apply in distributed systems and can be grounded in analytics-based decision making.

10.5.1 Norms for Trust

The foremost idea underlying trust is that the extent to which one party, Alice, may justifiably trust another party, Bob, depends on how Alice and Bob interact, including what Alice observes regarding Bob. I begin from two key points about trust.

Autonomy. Trust carries a connotation of choice: trust is meaningful only when the trusting party or *truster* can choose to proceed or not with the given interaction (with its counterparty, the *trustee*). And, the trustee can choose whether to carry out the interaction in question with the requisite quality. The latter element is diminished in cases of instrumental trust, wherein the truster treats the trustee as an instrument—as part of the infrastructure—and thus lacking in autonomy. In such cases, the question of trust reduces to the truster's trust in the reliability of the trustee.

Exposure. Trust carries a connotation of vulnerability: it is meaningful only when the trusting party has some skin in the game. If we were to somehow guarantee full protection to the trusting party, it could decide independently of its trust in any counterparty.

Given the foregoing points, I preserve the basic intuition that Alice's trust in Bob is strengthened when Bob meets or exceeds her expectations and weakened otherwise. Alice could observe Bob from a distance, as he interacts with others, but her learning of him would be the most relevant in circumstances where Alice personally faces a certain vulnerability to Bob's potential malice (or, in the case of an instrument, incompetence)—that is, when Alice has placed trust in Bob.

In departure from traditional research on trust, I express the relevant expectations formally in terms of *normative relationships* between Alice, Bob, and the other parties in the given system. From efforts in modelling large-scale systems from the standpoints of contractual relationships and decentralised administration, I have identified a small set of norm types [69]. Specifically, is Bob authorised to do something, committed to doing it, or prohibited from doing it? To what extent are Alice and Bob encounters regimented (so as to prevent violation) by their environment? To what extent is Alice directly protected? To what extent is Alice indirectly protected: would Bob be sanctioned (punished) for a violation? The lower the regimentation or protection the greater is Alice's vulnerability. Based on these, we can analyse Bob's actions from Alice's viewpoint—and Alice's actions from Bob's. Thus for each party we can provide a basis for determining how much trust to place in the other.

The norms I propose can all be expressed in conditional logic, e.g., [67]. Their logical basis offers a clear way to assign semantics to trust based on sets of possible computation paths [68]. We can use the semantics as a standard of correctness for any formal trust calculus, including as a basis for analytics. For example, Alice ought to trust Bob to no greater an extent for keeping his commitment to do P and Q than for keeping his commitment to do P alone. Such semantics can provide a basis for a rich variety of trust calculi that may be specialised for particular kinds of distributed software systems. Thus, if Alice determines (for example, via data mining) that Bob is trustworthy (to a specific extent) for P and Q, she should infer that Bob is at least as trustworthy for P. A potential practical benefit of the proposed approach is dealing with heterogeneous observations from which trust needs to be determined in the field. Such observations often do not respect simple patterns (such as being clear-cut positive or negative about the same P) from which one can compute a probability. I conjecture that a normative approach can provide a basis for extracting the most information from the observations that arise in practice.

Initially, we might pursue a Bayesian approach, which relates well both to analytics and to a semantics based on computation paths. In subsequent studies, it would be worthwhile to consider richer representations such as those based on utility theory.

Any approach to trust can be difficult to evaluate, and especially so if we bring in sophisticated concepts such as norms. Existing public datasets, e.g., for social networks, are limited and do not specify interactions. Indeed, the limited nature of available datasets is one of the important reasons for the popularity of the simplistic analytics-based approaches to trust. The following are two potential evaluation approaches.

- Obtain a text-rich dataset of user comments on each other and carry out text mining to infer assessments of the implicit norms involved and the felicity of the interactions involved.
- Develop a new dataset based on one or more games that we can have users play against each other. Such a dataset would likely be small but would point the way toward richer modelling.

10.5.2 A Call to Arms

To summarise, we see today a significant gap between trust theory and practice. Analytical and distributed systems approaches to trust involve shallow models not representative of real applications; even when these approaches talk of social networks, they do so in a manner that disregards any meaningful characterisation of the underlying social relationships. The cognitive models are richer but incomplete, yet difficult to characterise empirically from practically obtainable data.

The proposed norm-based research program will contribute a principled approach to trust that provides includes semantically rich abstractions for tasks and social or organisational relationships, yet which can be grounded in analytics-based decision making. This program is ambitious: it doesn't seek incremental improvements to current approaches, but to introduce a sea change in how trust is approached in research and practice, beginning from a foundation in norms.

Acknowledgments

Munindar Singh's effort was partially supported by the U.S. Army Research Office under grant W911NF-08-1-0105. The content of this paper does not necessarily reflect the position or policy of the U.S. Government; no official endorsement should be inferred or implied.

Nicoletta Fornara's effort is supported by the Hasler Foundation project nr. 11115-KG and by the SER project nr. C08.0114 within the COST Action IC0801 Agreement Technologies.

Henrique Lopes Cardoso's effort is supported by Fundação para a Ciência e a Tecnologia (FCT), under project PTDC/EIA-EIA/104420/2008.

Pablo Noriega's effort has been partially supported by the Spanish Ministry of Science and Technology through the Agreement Technologies CONSOLIDER project under contract CSD2007-0022, and the Generalitat of Catalunya grant 2009-SGR-1434.

References

1 Alexander Artikis. Dynamic protocols for open agent systems. In *Proceedings of the 8th International Conference on Autonomous Agents and Multiagent Systems - Volume 1*, AA-

MAS, pages 97–104, Richland, SC, 2009. International Foundation for Autonomous Agents and Multiagent Systems.

2 Gordon Baxter and Ian Sommerville. Socio-technical systems: From design methods to systems engineering. *Interacting with Computers*, 23(1):4–17, 2011.

3 G. Boella and L. van der Torre. Regulative and Constitutive Norms in Normative Multiagent Systems. In D. Dubois, C.A. Welty, and M.-A. Williams, editors, *Ninth International Conference on Principles of Knowledge Representation and Reasoning*, pages 255–266. AAAI Press, Whistler, Canada, 2004.

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

5 Jan Broersen. Issues in designing logical models for norm change. In George Vouros, Alexander Artikis, Kostas Stathis, and Jeremy Pitt, editors, *Organized Adaption in Multi-Agent Systems*, volume 5368 of *Lecture Notes in Computer Science*, pages 1–17. Springer Berlin Heidelberg, 2009.

6 Stefano Bromuri and Kostas Stathis. Distributed agent environments in the ambient event calculus. In *Proceedings of the Third ACM International Conference on Distributed Event-Based Systems*, DEBS '09, pages 1–12, New York, NY, USA, 2009. ACM.

7 J. Campos, M. López-Sánchez, J. A. Rodríguez-Aguilar, and M. Esteva. Formalising situatedness and adaptation in electronic institutions. In J. Hubner, E. Matson, O. Boissier, and V. Dignum, editors, *Coordination, Organizations, Institutions, and Norms in Agent Systems IV*, LNAI 5428, pages 126–139. Springer, 2009.

8 Marco Carbone, Mogens Nielsen, and Vladimiro Sassone. Formal model for trust in dynamic networks. In *Proceedings of the 1st International Conference on Software Engineering and Formal Methods (SEFM)*, pages 54–63. IEEE Computer Society, 2003.

9 J. Carmo and A. Jones. Deontic logic and contrary-to-duties. *Handbook of philosophical logic*, 8:265–343, 2002.

10 Cristiano Castelfranchi and Rino Falcone. *Trust Theory: A Socio-Cognitive and Computational Model.* Agent Technology. John Wiley & Sons, Chichester, UK, 2010.

11 Cristiano Castelfranchi, Rino Falcone, and Francesca Marzo. Being trusted in a social network: Trust as relational capital. In *Trust Management: Proceedings of the iTrust Workshop*, volume 3986 of *LNCS*, pages 19–32, Berlin, 2006. Springer.

12 Henry W. Chesbrough. *Open Innovation: The New Imperative for Creating and Profiting from Technology.* Harvard Business School Press, Boston, MA, 2003.

13 R.M. Chisholm. Contrary-to-duty imperatives and deontic logic. *Analysis*, pages 33–36, 1963.

14 Amit K. Chopra and Paolo Giorgini. Requirements engineering for social applications. In *Proceedings of the 5th International i* Workshop*, volume 766 of *CEUR Workshop Proceedings.* CEUR-WS.org, 2011. 138–143.

15 Amit K. Chopra and Munindar P. Singh. Multiagent commitment alignment. In *Proceedings of the Eighth International Conference on Autonomous Agents and MultiAgent Systems*, pages 937–944, 2009.

16 O. Cliffe, M. De Vos, and J. Padget. Specifying and reasoning about multiple institutions. In P. Noriega, J. Vázquez-Salceda, G. Boella, O. Boissier, V. Dignum, N. Fornara, and E. Matson, editors, *Coordination, Organizations, Institutions, and Norms in Agent Systems II*, LNAI 4386, pages 67–85. Springer, 2007.

17 S.E.S. Crawford and E. Ostrom. A grammar of institutions. *American Political Science Review*, pages 582–600, 1995.

18 V. Dignum, J. Vazquez-Salceda, and F. Dignum. Omni: Introducing social structure, norms and ontologies into agent organizations. In *Programming Multi-Agent Systems, Second International Workshop ProMAS 2004*, volume 3346 of *Lecture Notes in Artificial Intelligence (Subseries of Lecture Notes in Computer Science)*, pages 181–198, New York, NY, United States, 2005. Springer Verlag, Heidelberg, D-69121, Germany.

19 Mark d'Inverno, Michael Luck, Pablo Noriega, Juan A. Rodriguez-Aguilar, and Carles Sierra. Communicating open systems. *Artificial Intelligence*, 186(0):38 – 94, 2012.

20 Carl M. Ellison. Establishing identity without certificate authorities. In *Proceedings of the 6th USENIX Security Symposium*, pages 67–76, 1996.

21 M. Esteva, B. Rosell, J. A. Rodríguez-Aguilar, and J. L. Arcos. Ameli: An agent-based middleware for electronic institutions. In *Third International Joint Conference on Autonomous Agents and Multiagent Systems*, volume 1, pages 236–243, New York, USA, 2004. IEEE Computer Society.

22 Marc Esteva, Juan A. Rodriguez-Aguilar, Josep Lluis Arcos, and Carles Sierra. Socially-aware lightweight coordination infrastructures. In *AAMAS'11 12th International Workshop on Agent-Oriented Software Engineering*, pages 117–128, 2011.

23 Fernando Flores, Michael Graves, Brad Hartfield, and Terry Winograd. Computer systems and the design of organizational interaction. *ACM Transactions on Information Systems*, 6:153–172, 1988.

24 N. Fornara and M. Colombetti. Specifying and enforcing norms in artificial institutions. In G. Boella, L. van der Torre, and H. Verhagen, editors, *Normative Multi-agent Systems*, volume 07122 of *Dagstuhl Seminar Proceedings*. Schloss Dagstuhl, 2007.

25 N. Fornara and M. Colombetti. Specifying artificial institutions in the event calculus. In V. Dignum, editor, *Handbook of Research on Multi-Agent Systems: Semantics and Dynamics of Organizational Models*, Information science reference, chapter XIV, pages 335–366. IGI Global, 2009.

26 Nicoletta Fornara. Specifying and monitoring obligations in open multiagent systems using semantic web technology. In A. Elçi, M. Tadiou Kone, and M. A. Orgun, editors, *Semantic Agent Systems: Foundations and Applications*, volume 344 of *Studies in Computational Intelligence*, chapter 2, pages 25–46. Springer-Verlag, 2011.

27 Nicoletta Fornara and Marco Colombetti. Operational specification of a commitment-based agent communication language. In *Proceedings of the first international joint conference on Autonomous agents and multiagent systems: part 2*, AAMAS '02, pages 536–542, New York, NY, USA, 2002. ACM.

28 Nicoletta Fornara and Marco Colombetti. Representation and monitoring of commitments and norms using owl. *AI Commun.*, 23(4):341–356, 2010.

29 Nicoletta Fornara and Charalampos Tampitsikas. Using OWL Artificial Institutions for dynamically creating Open Spaces of Interaction. In *Proceedings of the AT 2012 First International Conference on Agreement Technologies, October 15 - 16, 2012 in Dubrovnik, Croatia*, volume 918 of *CEUR Workshop Proceedings*, pages 281–295, 2012.

30 Karen Fullam and K. Suzanne Barber. Dynamically learning sources of trust information: Experience vs. reputation. In *Proceedings of the 6th International Joint Conference on Autonomous Agents and MultiAgent Systems (AAMAS)*, pages 1062–1069, Honolulu, May 2007. IFAAMAS.

31 Andres Garca-Camino, Pablo Noriega, and Juan A. Rodrguez-Aguilar. Implementing norms in electronic institutions. In Simon Thompson Michal Pechoucek, Donald Steiner, editor, *Proceedings of the 4th International Joint Conference on Autonomous Agents and Multiagent Systems (AAMAS '05)*, pages 667–673, Utrecht, NL, 2005. ACM Press.

32 Andres Garcia-Camino, Juan A. Rodriguez-Aguilar, and Wamberto Vasconcelos. A distributed architecture for norm management in multi-agent systems. In Jaime Sichman,

Julian Padget, Sascha Ossowski, and Pablo Noriega, editors, *Coordination, Organization, Institutions and Norms in agent systems III*, volume 4870 of *Lecture Notes in Computer Science*, pages 275–286. Springer Berlin / Heidelberg, 2008.

33 Joseph Goguen. Requirements engineering as the reconciliation of technical and social issues. In M. Jirotka and J. Goguen, editors, *Requirements Engineering: Social and Technical Issues*, pages 165–200. Academic Press, 1994.

34 G. Governatori and A. Rotolo. Logic of violations: A gentzen system for reasoning with contrary-to-duty obligations. *Australasian Journal of Logic*, 4:193–215, 2006.

35 Guido Governatori, Zoran Milosevic, and Shazia Sadiq. Compliance checking between business processes and business contracts. In *Proceedings of the Tenth International Distributed Object Computing Conference*, pages 221–232, 2006.

36 Guido Governatori and Antonino Rotolo. Changing legal systems: Abrogation and annulment. part ii: Temporalised defeasible logic. In Guido Boella, Gabriella Pigozzi, Munindar P. Singh, and Harko Verhagen, editors, *NORMAS*, pages 112–127, 2008.

37 D. Grossi, H. Aldewereld, and F. Dignum. *Ubi Lex, Ibi Poena*: Designing norm enforcement in e-institutions. In P. Noriega, J. Vázquez-Salceda, G. Boella, O. Boissier, V. Dignum, N. Fornara, and E. Matson, editors, *Coordination, Organizations, Institutions, and Norms in Agent Systems II*, volume LNAI 4386, pages 101–114. Springer, 2007.

38 D. Grossi, D. Gabbay, and L. Torre. The norm implementation problem in normative multi-agent systems. In Mehdi Dastani, Koen V. Hindriks, and John-Jules Charles Meyer, editors, *Specification and Verification of Multi-agent Systems*, pages 195–224. Springer US, 2010.

39 Pascal Hitzler, Markus Krötzsch, and Sebastian Rudolph. *Foundations of Semantic Web Technologies*. Chapman & Hall/CRC, 2009.

40 J. Hübner, O. Boissier, R. Kitio, and A. Ricci. Instrumenting multi-agent organisations with organisational artifacts and agents. *Autonomous Agents and Multi-Agent Systems*, 20:369–400, 2010. 10.1007/s10458-009-9084-y.

41 Nicholas R. Jennings. On agent-based software engineering. *Artificial intelligence*, 117(2):277–296, 2000.

42 Andrew J. I. Jones and Marek J. Sergot. On the characterisation of law and computer systems: The normative systems perspective. In John-Jules Ch. Meyer and Roel J. Wieringa, editors, *Deontic Logic in Computer Science: Normative System Specification*, Wiley Professional Computing, chapter 12, pages 275–307. John Wiley and Sons, Chichester, UK, 1993.

43 Andrew J. I. Jones and Marek J. Sergot. A Formal Characterisation of Institutionalised Power. *Logic Journal of the IGPL / Bulletin of the IGPL*, 4:427–443, 1996.

44 Audun Jøsang. A subjective metric of authentication. In *Proceedings of the 5th European Symposium on Research in Computer Security (ESORICS)*, volume 1485 of *LNCS*, pages 329–344, Louvain-la-Neuve, Belgium, 1998. Springer.

45 H. Lopes Cardoso and E. Oliveira. Electronic institutions for B2B: Dynamic normative environments. *Artificial Intelligence and Law*, 16(1):107–128, 2008.

46 H. Lopes Cardoso and E. Oliveira. Norm defeasibility in an institutional normative framework. In M. Ghallab, C.D. Spyropoulos, N. Fakotakis, and N. Avouris, editors, *Proceedings of the 18th European Conference on Artificial Intelligence (ECAI 2008)*, pages 468–472, Patras, Greece, 2008. IOS Press.

47 H. Lopes Cardoso and E. Oliveira. A context-based institutional normative environment. In J. Hubner, E. Matson, O. Boissier, and V. Dignum, editors, *Coordination, Organizations, Institutions, and Norms in Agent Systems IV*, LNAI 5428, pages 140–155. Springer, 2009.

48 H. Lopes Cardoso and E. Oliveira. Monitoring directed obligations with flexible deadlines: a rule-based approach. In M. Baldoni, J. Bentahar, J. Lloyd, and M. B. Van Riemsdijk,

editors, *Declarative Agent Languages and Technologies VII*, LNAI, pages 51–67. Springer, 2010.

49 H. Lopes Cardoso and E. Oliveira. Social control in a normative framework: An adaptive deterrence approach. *Web Intelligence and Agent Systems*, 9:363–375, December 2011.

50 Ebrahim (Abe) Mamdani and Jeremy Pitt. Responsible agent behavior: A distributed computing perspective. *IEEE Internet Computing*, 4(5):27–31, September 2000.

51 S. Modgil, N. Faci, F. Meneguzzi, N. Oren, S. Miles, and M. Luck. A framework for monitoring agent-based normative systems. In Decker, Sichman, Sierra, and Castelfranchi, editors, *Proc. of 8th Int. Conf. on Autonomous Agents and Multiagent Systems*, pages 153–160. IFAAMAS, Budapest, Hungary, 2009.

52 Javier Morales, Maite López-Sánchez, and Marc Esteva. Using experience to generate new regulations. In *Proceedings of the twenty-second International Joint Conference on Artificial Intelligence IJCAI'11*, pages 307–312, Barcelona, Spain, 16/07/2011 2011. AAAI Press, USA.

53 John Mylopoulos, Lawrence Chung, and Brian Nixon. Representing and using nonfunctional requirements: a process-oriented approach. *IEEE Transactions on Software Engineering*, 18(6):483 –497, 1992.

54 Pablo Noriega. *Agent-Mediated Auctions: The Fishmarket Metaphor. PhD thesis Universitat Autònoma de Barcelona, 1997.* Number 8 in IIIA Monograph Series. IIIA, 1999.

55 Douglass C. North. *Institutions, Institutional change and economic performance.* Cambridge University Press, 1990.

56 Andrea Omicini, Alessandro Ricci, Mirko Viroli, Cristiano Castelfranchi, and Luca Tummolini. Coordination artifacts: Environment-based coordination for intelligent agents. In *Proceedings of the Third International Joint Conference on Autonomous Agents and Multiagent Systems - Volume 1*, AAMAS '04, pages 286–293, Washington, DC, USA, 2004. IEEE Computer Society.

57 P. Pasquier, R. A. Flores, and B. Chaib-Draa. Modelling flexible social commitments and their enforcement. In M.-P. Gleizes, A. Omicini, and F. Zambonelli, editors, *Engineering Societies in the Agents World V*, volume 3451 of *Lecture Notes in Artificial Intelligence*, pages 139–151. Springer, Toulouse, France, 2005.

58 Alessandro Ricci, Michele Piunti, and Mirko Viroli. Environment programming in multi-agent systems: an artifact-based perspective. *Autonomous Agents and Multi-Agent Systems*, 23(2):158–192, September 2011.

59 David Robertson. A lightweight coordination calculus for agent systems. In *Declarative Agent Languages and Technologies. DALT 2004*, volume 3476, pages 183–197. Springer, 2005.

60 William N. Robinson and Sandeep Purao. Specifying and monitoring interactions and commitments in open business processes. *IEEE Software*, 26(2):72–79, 2009.

61 Michael Schumacher and Sascha Ossowski. The governing environment. In Danny Weyns, H. Van Dyke Parunak, and Fabien Michel, editors, *Environments for Multi-Agent Systems II*, volume 3830 of *Lecture Notes in Computer Science*, pages 88–104. Springer Berlin / Heidelberg, 2006.

62 John R. Searle. *The Construction of Social Reality.* Free Press, New York, 1995.

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

64 Alberto Siena, Giampaolo Armellin, Gianluca Mameli, John Mylopoulos, Anna Perini, and Angelo Susi. Establishing regulatory compliance for information system requirements: An experience report from the health care domain. In *Proceedings of the 29th International Conference on Conceptual Modeling*, volume 6412 of *LNCS*, pages 90–103. Springer, 2010.

65 Herbert A. Simon. *Reason in Human Affairs.* Stanford University Press, 1983.

66 Munindar P. Singh. An ontology for commitments in multiagent systems: Toward a unification of normative concepts. *Artificial Intelligence and Law*, 7:97–113, 1999.

67 Munindar P. Singh. Semantical considerations on dialectical and practical commitments. In *Proceedings of the 23rd Conference on Artificial Intelligence (AAAI)*, pages 176–181, Chicago, July 2008. AAAI Press.

68 Munindar P. Singh. Trust as dependence: A logical approach. In *Proceedings of the 10th International Conference on Autonomous Agents and MultiAgent Systems (AAMAS)*, pages 863–870, Taipei, May 2011. IFAAMAS.

69 Munindar P. Singh. Norms as a basis for governing sociotechnical systems. *ACM Transactions on Intelligent Systems and Technology (TIST)*, pages 1–21, 2013. To appear; available at `http://www.csc.ncsu.edu/faculty/mpsingh/papers`.

70 Munindar P. Singh and Amit K. Chopra. Correctness properties for multiagent systems. In *Proceedings of the Sixth Workshop on Declarative Agent Languages and Technologies*, volume 5948 of *LNCS*, pages 192–207. Springer, 2009.

71 Munindar P. Singh, Amit K. Chopra, and Nirmit Desai. Commitment-based service-oriented architecture. *IEEE Computer*, 42(11):72–79, 2009.

72 C. Tampitsikas, S. Bromuri, and M. Schumacher. Manet: A model for first-class electronic institutions. In S. Cranefield and P. Noriega, editors, *12th International Workshop on Coordination, Organization, Institutions and Norms in Agent Systems (COIN)*, pages 105–119, Taipei, Taiwan, 2011.

73 Charalampos Tampitsikas, Stefano Bromuri, Nicoletta Fornara, and Michael Ignaz Schumacher. Interdependent Artificial Institutions In Agent Environments. *Applied Artificial Intelligence*, 26(4):398–427, 2012.

74 Nick Tinnemeier, Mehdi Dastani, and John-Jules Meyer. Programming norm change. In *Proceedings of the 9th International Conference on Autonomous Agents and Multiagent Systems: volume 1*, AAMAS, pages 957–964, Richland, SC, 2010. International Foundation for Autonomous Agents and Multiagent Systems.

75 Y. B. Udupi and M. P. Singh. Dynamics of contracts-based organizations: A formal approach based on institutions. In *Proceedings of the 6th international joint conference on Autonomous agents and multiagent systems*, AAMAS, pages 19:1–19:3, New York, NY, USA, 2007. ACM.

76 J. Vázquez-Salceda, H. Aldewereld, and F. Dignum. Implementing norms in multiagent systems. In G. Lindemann, J. Denzinger, I. J. Timm, and R. Unland, editors, *Multiagent System Technologies*, volume 3187 of *Lecture Notes in Artificial Intelligence*, pages 313–327, Erfurt, Germany, 2004. Springer Verlag, Heidelberg, D-69121, Germany.

77 Javier Vázquez-Salceda, Virginia Dignum, and Frank Dignum. Organizing multiagent systems. *Autonomous Agents and Multi-Agent Systems*, 11:307–360, 2005.

78 Mahadevan Venkatraman and Munindar P. Singh. Verifying compliance with commitment protocols: Enabling open Web-based multiagent systems. *Autonomous Agents and Multi-Agent Systems*, 2(3):217–236, September 1999.

79 H. J. E. Verhagen. *Norm Autonomous Agents*. PhD thesis, The Royal Institute of Technology and Stockholm University, 2000.

80 Yonghong Wang and Munindar P. Singh. Formal trust model for multiagent systems. In *Proceedings of the 20th International Joint Conference on Artificial Intelligence (IJCAI)*, pages 1551–1556, Hyderabad, 2007. IJCAI.

81 Danny Weyns, Andrea Omicini, and James Odell. Environment as a first class abstraction in multiagent systems. *Autonomous Agents and Multi-Agent Systems*, 14(1):5–30, 2007.

82 Pınar Yolum and Munindar P. Singh. Flexible protocol specification and execution: Applying event calculus planning using commitments. In *Proceedings of the 1st International*

Joint Conference on Autonomous Agents and MultiAgent Systems, pages 527–534. ACM Press, 2002.

83 Eric S.K. Yu. Towards modelling and reasoning support for early-phase requirements engineering. In *Proceedings of the Third IEEE International Symposium on Requirements Engineering*, pages 226–235, 1997.

84 Franco Zambonelli, Nicholas R. Jennings, and Michael Wooldridge. Developing multiagent systems: The Gaia methodology. *ACM Transactions on Software Engineering Methodology*, 12(3):317–370, 2003.