Electronic Institutions with Normative Environments for Agent-based E-contracting

Henrique Lopes Cardoso

Universidade do Porto
Faculdade de Engenharia
FEUP

Faculty of Engineering, University of Porto
Department of Informatics Engineering

September 2010

Electronic Institutions with Normative Environments for Agent-based E-contracting

Henrique Lopes Cardoso

Universidade do Porto
Faculdade de Engenharia
FEUP

Supervisor: Professor Doutor Eugénio Oliveira

Thesis submitted in partial fulfillment of the requirements for the Doctoral degree in Informatics Engineering.

This work was supported by Fundação para a Ciência e a Tecnologia (FCT) through grant SFRH/BD/29773/2006.

To my *Parents*, who taught me my first steps,

to *Ró*, who walks besides me,

and to our beloved offspring, *Luísa* and *Marta*.

Acknowledgments

This has been a long journey. Sometimes wondering, sometimes running, sometimes walking, I think I managed to accomplish something. For that I was fortunate enough to be surrounded by some splendid persons.

Family comes first. I am infinitely grateful to my parents, who have always showed me their unconditional love and support. Doing a PhD research work inevitably interferes with one's personal life. I thank my wife, for letting me know that I am most loved and needed at home, and for understanding my mood changes or mind absence in those days where my mind was either blank-like or bubbling in ideas.

Someone said that while in a PhD venture you should keep life changes (such as getting married, moving house or ordering kids) as low as possible. Among other far less important events, since the beginning of my research I became a father of two wonderful daughters. I can't imagine my life without them. I thank them for their joy and for all the good nights' sleep they allowed me.

Now let's back to work. I am deeply thankful to Prof. Eugénio Oliveira, my mentor and principal enthusiast of the Electronic Institution project. Despite his overfilled schedule, he has always found the time to brainstorm with me and come up with invaluable ideas, some of which are included in this thesis. I also thank him for "keeping the carrot out of reach", which has made me evolve in a way that perhaps only sometime in the future I will fully understand.

Some words for my esteemed colleagues at LIACC-NIAD&R. I would like to thank Ana Paula Rocha and Joana Urbano, whose research work is also part of the project I have been working in. Many fruitful discussions have taken place and many more are to come. Luís Paulo Reis has always shown his respect for my research, and is a great faculty colleague. António Castro and António Pereira have been great in reminding me that there is (lab) life beyond research. Chats at coffee breaks are as important as research work. Luís Sarmento has shown a great deal of altruism in promoting a reading group for enhancing research collaboration. Unfortunately I was not sensible enough to make a stronger contribution.

A word of appreciation goes to my advisory group. I am grateful to Prof. Luís Camarinha-Matos and to Ana Paula Rocha, for reading and giving me feedback both on my thesis proposal and on a first draft of this document. In this respect, I would also like to thank Prof. Virginia Dignum and Prof. Frank Dignum.

Throughout this research work I had many of those essential moments when a researcher presents his contributions to the research community. In some of those moments I was fortunate to come across Prof. Pablo Noriega who, without any duty to do so, has always encouraged me with his comments. Thank you.

Finally, I would like to thank Fundação para a Ciência e a Tecnologia (FCT) for the doctoral degree grant SFRH/BD/29773/2006. Additionally, this research has been partially supported by project POSC/EIA/57672/2004.

Abstract

The regulation of interactions in open multi-agent systems is an increasingly relevant research topic. The field of *normative multi-agent systems* looks at such interactions as norm-governed: agent behaviors are guided by norms, dictating what they should do or do not in particular situations. *Electronic institutions*, inspired on real-world institutions that regulate interactions between members of a society, have been studied and developed as a means of delivering regulated (artificial) multi-agent environments.

Running business-to-business relationships over the Internet is quickly becoming the standard way of doing business. While technological advances are mainly concerned with matchmaking and business integration from an implementation point of view, *electronic contracting*, seen as the set of activities that allow companies to electronically establish and enact contractual relations, has still a long way to go. Nevertheless, the automation of electronic contracting is acknowledged by some researchers as highly beneficial to companies.

In this thesis we investigate on the role electronic institutions can play in the domain of agent-based electronic contracting. An *Electronic Institution (EI)* is here seen as a computational infrastructure that assists software agents in the process of creating and running contractual relationships. For that purpose, in our view the EI includes a set of services focusing on the automation of activities such as contract negotiation, monitoring and enforcement. The main subject of this thesis concerns the specification, development and exploitation of an *institutional normative environment* for electronic contracting.

We propose a new formalization of *contractual obligations*, based on temporal logic. Contractual obligations are seen as directed obligations with flexible time windows. Our contribution allows agents to reason explicitly about temporal violation states, and enables counterparties of obligations to participate in the monitoring process.

Electronic contracts, obtained through a process of negotiation, are added to the normative environment for monitoring and enforcement purposes. This environment includes a *hierarchical normative framework* intended to assist contract establishment, whose specification is a major contribution of this thesis. By ex-

hibiting a *norm defeasibility* model, the normative framework is adaptable and extensible to different contracting situations. We present a *rule-based* implementation of a *contract monitoring service*, and validate its correctness by defining several contracting scenarios. The monitoring process is then applied to a number of possible contract enactments showing different outcomes. This implemented prototype, embedded in a more comprehensive Electronic Institution Platform, is another contribution of this thesis.

A final contribution is an initial prospect towards norm enforcement within the institutional environment, based on an *adaptive deterrence sanctioning* model that tries to enforce norm compliance without excessively compromising agents' willingness to establish contracts. We propose a novel abstract contract representation, based on the notion of *commitment trees*. In an agent population, we characterize agents using two main parameters: *risk tolerance* and *social awareness*. Using a social simulation approach, we show that, according to our proposed adaptive deterrence model, deterrence sanctions tend to be less needed when agents are more socially concerned, which is in accordance with real-life experience.

Resumo

A regulação das interacções em sistemas multi-agentes abertos é um tema de investigação de relevância crescente. A área dos *sistemas multi-agentes normativos* considera tais interacções como sendo governadas por normas: os comportamentos dos agentes são guiados por normas, indicando o que eles devem ou não fazer em situações concretas. As *instituições electrónicas*, inspiradas em instituições do mundo real que regulam interacções entre os membros de uma sociedade, têm sido estudadas e desenvolvidas como uma forma de obter ambientes (artificiais) multi-agentes regulados.

A condução de relações do tipo *business-to-business* na Internet está rapidamente a tornar-se na forma *standard* de efectuar negócios. Enquanto que os avanços tecnológicos se debruçam principalmente sobre a pesquisa de parceiros potenciais e sobre a integração do negócio de um ponto de vista implementacional, a *contratação electrónica*, vista como o conjunto de actividades que permitem às empresas estabelecer e conduzir relações contratuais de forma electrónica, tem ainda um longo caminho a percorrer. No entanto, a automação da contratação electrónica é reconhecida por alguns investigadores como sendo altamente benéfica para as empresas.

Nesta tese investigamos o papel que as instituições electrónicas podem desempenhar no domínio da contratação electrónica baseada em agentes computacionais. Uma *Instituição Electrónica (EI)* é aqui vista como uma infra-estrutura computacional que assiste agentes de software no processo de criação e execução de relações contratuais. Para tal, na nossa perspectiva a EI inclui um conjunto de serviços virados para a automação de actividades tais como a negociação de contratos, sua monitorização e coerção. O assunto principal desta tese diz respeito à especificação, desenvolvimento e exploração de um *ambiente normativo institucional* para contratação electrónica.

Propomos uma nova formalização de *obrigações contratuais*, baseada em lógica temporal. As obrigações contratuais são vistas como obrigações direccionadas com janelas temporais flexíveis. A nossa contribuição faz com que os agentes possam raciocinar explicitamente sobre estados de violação temporal, e permite à contraparte de uma obrigação participar no processo de monitorização.

Os contratos electrónicos, obtidos por um processo de negociação, são adicionados ao ambiente normativo com a finalidade da sua monitorização e coerção. Este ambiente inclui um *enquadramento normativo hierárquico* cujo objectivo é assistir a criação de contratos, e cuja especificação é uma contribuição principal desta tese. Ao exibir um modelo de *revogação de normas*, o enquadramento normativo é adaptável e extensível a diferentes situações contratuais. Apresentamos uma implementação *baseada em regras* de um *serviço de monitorização de contratos*, e validamos a sua correcção definindo diversos cenários contratuais. O processo de monitorização é depois aplicado a um conjunto de possíveis execuções de contrato com diferentes resultados. Este protótipo, implementado e integrado numa Plataforma Instituição Electrónica mais abrangente, é outra contribuição desta tese.

Uma contribuição final é uma abordagem inicial à problemática da coerção de normas no seio de um ambiente institucional, baseada num modelo *adaptativo de sanções dissuasoras* que tenta impor o cumprimento de normas sem comprometer excessivamente a predisposição dos agentes em estabelecer contratos. Propomos uma nova representação abstracta de contrato, baseada na noção de *árvore de compromissos*. Numa população de agentes, caracterizamos os agentes com base em dois parâmetros principais: *tolerância ao risco* e *consciência social*. Recorrendo a uma abordagem de simulação social, mostramos que, de acordo com o nosso modelo adaptativo de dissuasão, as sanções tendem a ser menos necessárias quando os agentes exibem um comportamento socialmente mais aceitável, o que vai de encontro à experiência do mundo real.

Résumé

La régulation des interactions dans les systèmes multi-agents ouvert est un sujet de recherche de plus en plus pertinent. Le domaine des *systèmes multi-agents normatifs* regarde telles interactions comme gouvernées par normes: les comportements des agents sont guidés par des normes, qui indique ce qu'ils doivent ou doivent pas faire en situations particulières. Les *institutions électroniques*, inspirés sur les institutions du monde réel qui régissent les interactions entre les membres dune société, ont été étudiés et développés comme un moyen de fournir des environnements (artificielles) multi-agents régulées.

La mise en marche des relations *business-to-business* sur l'Internet est rapidement devenu le moyen standard de conduire des affaires. Bien que les avancées technologiques sont principalement concernées par la cherché de potentiel partenaires et l'intégration des entreprises d'un point de vue de l'implémentation, la *contractualisation électronique*, considéré comme l'ensemble des activités qui permettent aux entreprises d'établir et d'exécuter, par voie électronique, des relations contractuelles, a encore un long chemin à parcourir. Néanmoins, l'automatisation de la contractualisation électronique est reconnu par certains chercheurs comme très avantageux pour les entreprises.

Dans cette thèse, nous étudions le rôle que les institutions électroniques peuvent jouer dans le domaine de la contractualisation électronique à base d'agents. Une *Institution Électronique (EI)* est ici considéré comme une infrastructure computationnelle qui aide les agents logiciels dans le processus de création et exécution des relations contractuelles. À cet effet, à notre avis la EI comprend un ensemble de services axés sur l'automatisation des activités telles que la négociation des contrats, le suivi de son exécution et application. Le sujet principal de cette thèse concerne la spécification, le développement et l'exploitation d'un *environnement normatif institutionnel* pour la contractualisation électronique.

Nous proposons une nouvelle formalisation des *obligations contractuelles*, basé sur la logique temporelle. Les obligations contractuelles sont considérées comme des obligations directionnels avec fenêtres de temps flexibles. Notre contribution permet aux agents de raisonner explicitement sur des violations temporelles, et permet aux contreparties de l'obligation de participer au processus de suivi.

Des contrats électroniques, obtenu par un processus de négociation, sont ajoutés à l'environnement normatif à fin de surveillance et application. Cet environnement comprend un *cadre normatif hiérarchique* qui vise à aider à la création de contrats, dont la spécification est une contribution majeure de cette thèse. En présentant une modèle de *révocation des normes*, le cadre normatif est adaptable et extensible à différents situations contractuelles. Nous présentons une implémentation *basé à règles* d'un *service de surveillance du contrat*, et nous validons sa conformité avec la définition des différents scénarios contractuelles. Le processus de suivi est ensuite appliqué dans un certain nombre de possible exécutions de contrats avec différents résultats. Ce prototype, implémenté et intégré dans une plus large Plateforme Institution Électronique, constitue une autre contribution de cette thèse.

Une dernière contribution est un approche initiale vers l'application des normes au sein de l'environnement institutionnel, fondé sur un modèle *adaptative de sanctions de dissuasion* qui essaie de faire respecter les normes sans trop compromettre la volonté des agents à établir des contrats. Nous proposons une nouvelle représentation abstraite du contrat, fondée sur la notion de *arbre d'engagements*. Dans une population d'agents, nous caractérisons les agents au moyen de deux paramètres principaux: *tolérance au risque* et *conscience sociale*. En utilisant une approche de simulation sociale, nous montrons que, selon notre modèle adaptatif de dissuasion, les sanctions tendent à être moins nécessaire lorsque les agents sont socialement plus concernés, ce qui est conforme à l'expérience de la vie réelle.

Contents

List of Tables

List of Figures

Chapter 1

Introduction

Multi-agent systems (MAS) research provides an increasingly relevant perspective to artificial intelligence and to computing in general: that of modeling systems where the interaction among computing elements (software agents) is of paramount importance [152]. Researchers in this field are therefore no longer solely interested in building intelligent individual software agents, but also in designing suitable interactions among these agents and modeling organizations that they may be a part of.

This thesis is centered around the role of *electronic institutions*, understood as computational infrastructures, in assisting software agents in the process of creating and running contractual relationships. The development of autonomous software agents that are able to use these infrastructures is a daunting task of which we will only scratch the surface. In fact, the target domain that we address – business-to-business contracting – may pose too many obstacles to such an endeavor (both in terms of technological limitations and real-world acceptance). Throughout this thesis we will be more interested in the supporting features that can be developed to assist partially autonomous software agents in tasks related with the establishment and enactment of contracts regulating joint work. Partial autonomy is here used in the sense that, despite significant automation, human interaction is likely to be needed in several steps of the process.

In this chapter we provide motivation for this work, present the rationale behind the research done and summarize the main contributions that were achieved. We also outline the structure of the rest of this document.

1.1 The Evolution of Business

The shift, in the last decades, from an industrial economy (based on mass production models) to an information-based economy associated with the globalization of markets, has brought an enormous increase in competitiveness, leading to the need for new organizational models (see e.g. [73] for some insights on this issue). In order to prosper in such an environment, enterprises need to establish synergies through collaborative links, sometimes even with competitors: different enterprises coordinate the necessary means to accomplish shared activities or reach common goals. This association of strengths enables enterprises to build privileged relationships, based on an increase of advantages through resource and competence sharing, leading to risk minimization.

Cooperation arrangements are particularly relevant for small and medium enterprises (SME) [2], due to their reduced size, high specialization and flexibility. These kinds of enterprises have been adopting new strategies that enable them to adapt to a constantly changing market, by organizing themselves in strategic partnerships. While maintaining their business independence, partners are able to reach otherwise unreachable (physical and customer) markets and take advantage of economies of scale. Furthermore, many large companies are isolating parts of their businesses, making them autonomous and more agile, in order to increase the overall flexibility and achieve greater performances. Outsourcing models are also becoming dominant, enabling enterprises to concentrate on their core competencies. Thus, there is an increasing emphasis in temporary cooperation and coordination of small business units.

The concept of a Virtual Enterprise (VE) has also arisen from the trend towards coordination of several business entities, providing a more decentralized approach. As noted by Petersen [176], several definitions have been proposed for the VE concept, from within different research communities. One possible definition has been advanced by Camarinha-Matos [28]: "a temporary alliance of enterprises that come together to share skills or core competences and resources in order to better respond to business opportunities, and whose cooperation is supported by computer networks.". However, the establishment of cooperation agreements between enterprises is not a new phenomenon, rather belonging to the very nature of the business world; the use of communication and information technologies to support agile cooperation brings a new level of effectiveness and a substantial increase of cooperation on a global scale, as distance is no longer a major limiting issue. Other definitions highlight the fact that enterprises participating in a VE are independent and autonomous entities [76], or emphasize the temporary nature of a VE and the dynamism of the business environment in which they are to be created [55].

As a consequence of this evolution, there is a major trend on developing advanced software tools that support the establishment and management of cooperation agreements among companies.

1.2 Conducting Electronic Business

From a technological perspective, the advent of the Internet has brought to the business realm new ways of exchanging data between companies, assisted by computer communication networks. The first technological impact has thus been on changing the way companies exchange documents (e.g. through e-mails), which are turned into an electronic form. A pioneering standard that was created at this stage was EDI[1], which enabled companies to exchange structured messages that could be automatically interpreted by their information systems.

After information and communication technologies have matured, electronic commerce (*e-commerce* for short) has become a common term, as more and more companies join the "on-line" and electronic paradigm of doing business. While this is true for business-to-consumer relationships, e-commerce has a much stronger importance and is becoming everyday more widespread in the business-to-business (B2B) world. Companies have started to automate some of their business activities, namely the process of finding potential business partners. This means that besides holding a world-wide-web presence through websites, companies have created business portals that more powerfully allow them to manage their B2B relationships. Furthermore, third-party on-line business marketplaces (e.g. MFG.com[2]) have been created that allow SMEs to pursue business opportunities in the world-wide-web.

Also, a number of standards has grown in the e-commerce domain, defined by standardization bodies such as UN/CEFACT[3] or OASIS[4].

The ebXML[5] suite of specifications intends to enable enterprises of any size and geographical location to enter into business over the Internet, by specifying standard methods to exchange business messages, conduct trading relationships, communicate data in common terms and define and register business processes. Enterprises are seen as trading partners that register their business profiles in an easily accessible ebXML registry. After an initial discovery phase, trading partners define a business arrangement that outlines a mutually agreed upon business scenario (a collaboration protocol agreement), including messaging requirements for transactions to take place.

Another suite of specifications has been created by the Web Services Activity of the W3C[6], targeted more broadly at application-to-application communication. Web-service standards include specifications for the exchange of messages

[1]Electronic Data Interchange

[2]`http://www.mfg.com`

[3]United Nations Centre for Trade Facilitation and Electronic Business – `http://www.unece.org/cefact/`

[4]Organization for the Advancement of Structured Information Standards – `http://www.oasis-open.org`

[5]Electronic Business using eXtensible Markup Language – `http://www.ebxml.org`

[6]World Wide Web Consortium – `http://www.w3.org`

(SOAP[7]), for the description of services that are provided by applications and that can be invoked through SOAP messages (WSDL[8]), and for infrastructures required to publish and discover services at runtime (UDDI[9]).

A related standard has been promoted to foster the use of web-service technology in business applications: WS-BPEL[10] is an orchestration language for specifying business process behavior based on web-services. While the interaction model supported by WSDL is a stateless request-response model, WS-BPEL provides a means to define sequences of peer-to-peer web-service invocations that implement a business interaction protocol.

Several other standardization efforts have been pursuit, some of which are precursors of the above mentioned ones.

Time has now come for a more integrating approach on automated handling of (parts of) the contracting activity, ranging from partner matchmaking to contract establishment and to business enactment. This gives rise to electronic contracting (*e-contracting* for short), seen as a further step on using information and communication technologies in the core of a business relationship – the contract, which is to be handled in an electronic form (*e-contract*). However, as noted by Angelov [6] the development of automated tools for handling e-contracts presents many technological, business and legal challenges.

Standardization efforts on e-contracting have also been attempted. For instance, the LegalXML[11] initiative aims at creating standards for the electronic exchange of legal data. In particular, it has delivered the LegalXML eContracts specification, which aims at handling contract documents and terms. The LegalXML Technical Committee has considered the scenario of having machine readable contracts useful for performance monitoring and dispute resolution [157]. However, the specification has left out a deontic contract language approach, acknowledging that this is a specialized area that requires considerable further research before commercial adoption. Deontic issues are, nevertheless, of paramount importance in contractual relationships, and their explicit representation in some form is crucial to enable automatic contract monitoring.

A number of rule-based engines exist that allow for the specification of business rules, which can in turn be used to specify the deontic aspects of contracts, making them available for machine processing (e.g. for run-time compliance checking). A standardization effort in this domain is the Rule Markup Initiative[12], which attempts to define RuleML[13] as a common language representation for all kinds of rules (not only for business or e-contract purposes).

[7] Simple Object Access Protocol

[8] Web Services Description Language

[9] Universal Description, Discovery and Integration

[10] Web Services Business Process Execution Language

[11] `http://www.legalxml.org`

[12] `http://www.ruleml.org`

[13] Rule Markup Language

1.3 Research Rationale and Outputs

The purpose of this thesis is to investigate and make contributions regarding the development of infrastructures that support e-contracting. We employ a MAS research perspective to this problem.

There has been substantial research on the application of MAS to B2B (see e.g. the surveys [19] and [109]). A vast amount of approaches to B2B automation has concentrated on the design of market mechanisms (e.g. auctions) and negotiation protocols [121][91], which together with research on bidding and negotiation strategies comprise what might be called approaches to *agent-based automated negotiation*. Negotiation protocols and tools that allow for partner selection in VE scenarios have also been proposed (namely by Rocha [185] and Petersen [176]).

Comparatively less approaches focus on e-contract establishment, that is, on how software agents might be able to negotiate the *normative content* of e-contracts. By normative content we mean the business commitments or obligations that contractual partners commit to when establishing a contract. There are, however, several research initiatives concerned with capturing the normative aspects of e-contracts (many of which are covered in the next chapters).

The research we have done and report in this thesis concerns employing the concept of *electronic institutions* to the domain of agent-based B2B e-contracting. Real-world institutions are used to regulate the interactions between members of a society. An *Electronic Institution (EI)* is the electronic counterpart of one of such institutions, specifying and enforcing regulations on electronic members (software agents) that adhere to the underlying electronic society on behalf of real enterprises. There have been other researchers exploring and developing the notion of electronic institutions within the MAS research field (namely Noriega [166], F. Dignum [60], Rodríguez-Aguilar [188], Esteva [68], Vázquez-Salceda [226]). Our approach tries to provide a practical and open account to the EI concept, thinking of it as a coordination framework that assists agents on establishing contractual relationships that will be a part of the regulations guiding future interactions between such agents.

A subfield of MAS research that has grown in importance in the last years is the field of *normative multi-agent systems (NMAS)* [23]. A normative system has been defined as a "set of interacting agents whose behaviour can usefully be regarded as governed by norms. Norms prescribe how agents ought to behave, and specify how they are permitted to behave and what their rights are" [116]. Agents therefore interact in some kind of *normative environment* that shapes, or at least influences, their behavior options. We believe that this normative perspective on MAS is perfectly suited for the e-contracting domain, and we have used NMAS as the underlying scene for applying MAS in this domain.

1.3.1 Research questions

As mentioned above, the guiding line of this research is the application of electronic institutions and normative multi-agent systems to the domain of B2B e-contracting. Therefore, a major effort has been put on analyzing the EI concept from the point of view of its potential importance in enabling the use of software agents for automating e-contracting processes. In particular, we have tried to pursue the following research questions:

1. *How can the concept of Electronic Institution be adapted to address the domain of agent-based e-contracting?*

   Although there have been some approaches to design and develop the concept of Electronic Institution, they somehow lack the flexibility of addressing the creation of normative relationships (contracts) at run-time. We therefore need to rethink the concept in terms of its applicability to the e-contracting domain.

2. *What kind of e-contracting support should we embed in a normative environment?*

   Having NMAS as our underlying scene, and considering that e-contracts are created at run-time, we have a setting where interacting agents are willing to establish normative relationships. We need to investigate on the kinds of infrastructures that can be provided to assist these agents on their effort to establish contracts and to enact those contracts.

3. *How should e-contracts be specified in order to enable their automatic processing?*

   An important phase of e-contracting is the enactment of contracts. If an Electronic Institution is to be able to monitor contract enactment, e-contracts should be specified in a way that enables them to be interpreted in an automatic way. Taking a NMAS perspective, we need to study how norms can be used to model the normative content of e-contracts.

4. *Which mechanisms may an Electronic Institution, and more specifically a normative environment, put in practice to enforce e-contracts?*

   Taking agents as autonomous entities from the Electronic Institution's perspective, they are able to decide whether to fulfill or not their contractual duties. Therefore, it is important to consider extra-contractual enforcement mechanisms that can be put into place in order to influence agent behavior.

These questions are obviously interrelated. The specification of e-contracts influences and is dependent on the e-contracting infrastructures that are to be available. The "institutional" flavor of the Electronic Institution concept is closely related with its ability to monitor and enforce norms, which should be identified as an important ingredient of the infrastructures to develop.

1.3.2 Main contributions

This research has focused on building infrastructures for agent-based B2B e-contracting. The main scientific contributions are centered on the specification, development and exploitation of an institutional normative environment. We therefore highlight the following contributions:

- A specification of an *institutional normative environment*

  Software agents, representing different business entities, negotiate and establish contractual relationships. Such relationships are expressed by a set of contractual norms that will guide further interactions among contractual partners. The normative environment that we propose has the responsibility to monitor norm abidance, by maintaining a normative state and applying monitoring rules and contractual norms. Our approach is inspired by Searle's work on institutional reality [203], in order to bring facts into the normative environment. Also, the environment is institutional in the sense that it is empowered to enforce norms should they be violated.

- A *hierarchical normative framework* model with *norm defeasibility*

  In order to facilitate the establishment of e-contracts, the normative environment includes a supportive and extensible normative framework model. This model is inspired by notions coming from contract law theory, namely the use of "default rules" – background norms to be applied in the absence of any explicit agreement to the contrary. This normative structure is composed of a hierarchy of *contexts*, within which norms are created that may apply to sub-contexts. The context hierarchy tries to mimic the fact that in business it is often the case that a B2B contractual agreement forms the business context for more specific contracts that may be created. Each contract establishes a new context for norm applicability. A *norm defeasibility* approach is proposed in order to determine whether a norm should be inherited, for a specific situation, from an upper context. This feature allows the normative framework to be adapted (to better fit a particular contract case) and extended (allowing new contract types to be defined).

- A formalization of *contractual obligations* using temporal logic

  Our approach, taking inspiration from real-world legislations on trade contracts, sees contract enactment as a cooperative activity. The successful performance of business is supposed to benefit all involved parties, and therefore contractual partners may be cooperative enough to allow, up to a certain extent, counterparty deviations. The semantics of contractual obligations, as we model them, incorporate flexible livelines and deadlines, and include the possibility for agents to participate in the monitoring process.

- An implemented prototype (proof-of-concept) integrating the scientific contributions highlighted above

  An institutional normative environment including the aforementioned features has been implemented, based on a forward-chaining rule-based inference engine – Jess[14] [82]. Furthermore, this environment has been integrated into an *Electronic Institution Platform* also providing negotiation and contracting facilities.

- A preliminary model of *adaptive deterrence sanctions* for norm enforcement

  Adaptive enforcement mechanisms are important in open environments, where the behavior of an agent population cannot be directly controlled. When the normative specification of contracts has imperfections, namely by not specifying normative consequences for every possible contract enactment outcome, self-interested agents may try to exploit their potential advantage and intentionally violate contract clauses. We have designed and experimentally evaluated a model for adaptive deterrence sanctions that tries to enforce norm compliance without excessively compromising agents' willingness to establish contracts. Raising deterrence sanctions has a side effect of increasing the risk associated with contracting activities.

The research reported in this thesis is framed into a wider research project, taking place at LIACC[15] (in the NIAD&R[16] group) and under supervision of Prof. Eugénio Oliveira, which concerns the development of an Electronic Institution Platform for B2B contracting. Several agent-based services have been identified and developed that assist the automation of e-contracting processes, such as partner selection through automatic negotiation [172][185], ontology-mapping [155][154], contract establishment, monitoring and enforcement (including the work reported in this thesis), and computational trust and reputation [219]. These services have been integrated in a common software platform (based on JADE[17] [18] and using well-defined interaction protocols), and an effort has been put on aligning the operation of the normative environment with the outcome of the negotiation and contract establishment service. Contract drafting makes use of a particular contract model we have developed. The normative environment provides a contract monitoring and enforcement service, and has been integrated with subscription mechanisms in place for the notification of contract-related events to interested and authorized agents.

[14] `http://www.jessrules.com`

[15] Laboratório de Inteligência Artificial e Ciência de Computadores

[16] Núcleo de Inteligência Artificial Distribuída & Robótica

[17] Java Agent DEvelopment Framework – `http://jade.tilab.com`

1.4 Structure of this Thesis

The rest of this thesis is divided into two parts.

Part I provides the reader a perspective on research that has been carried out in two research communities devoted to distinct but somehow related research areas. The work presented in this thesis can be seen as an effort to bring these research areas closer to each other: *electronic contracting* and *normative multi-agent systems*.

- **Chapter 2** provides a background overview on the issue of electronic contracting. We start by dissecting the stages that may be comprised in this activity and describe the key issues in each of those stages. When doing so, we refer to numerous approaches of other researches dealing with such issues. The chapter also brings to light a number of relevant research projects that are related to electronic contracting.

- **Chapter 3** delves into the research realm of normative multi-agent systems. We start by an introduction to norms and their use within the MAS research community. We then provide an overview of some research perspectives on providing regulated environments for MAS. The central concept of this thesis – Electronic Institution – is then studied in detail as a means for delivering a regulated multi-agent environment. The notion of institution is first presented from a multidisciplinary perspective. The most relevant approaches to conceptualize, design or develop institutions in MAS, from different research groups, are discussed.

Part II includes the research contributions of this thesis, while trying to answer the research questions raised in Section 1.3.1.

- **Chapter 4** presents the Electronic Institution concept from the LIACC-NIAD&R perspective, with an e-contracting application in mind. Some of the services to be provided by an Electronic Institution Platform are presented, even though they are outside the scope of this thesis. We also shed some light on the kinds of contractual scenarios that we aim at with the research developments contained in this thesis. Chapter 4 answers the first research question.

- **Chapter 5** provides a specification for an institutional normative environment, including a norm representation formalism. The normative environment provides an e-contract monitoring and enforcement facility, by applying monitoring rules and contractual norms. It also includes a context-based hierarchical normative framework that agents can exploit when creating their own e-contracts. Chapter 5 answers the second research question, and partially the third.

- **Chapter 6** proposes a formalization of contractual obligations using temporal logic. This formalization tries to take into account the way contracts are enacted in the real-world, taking inspiration from legislations on trade contracts. The semantics of contractual obligations is modeled with a flexible approach to temporal violations, identifying the bearer and counterparty roles in a directed obligation and enabling enacting agents to participate in the monitoring process. Chapter 6 provides a reinforcement to answer the third research question.

- **Chapter 7** explains how the ideas presented in the preceding chapters have been implemented using a rule-based inference engine. It also illustrates the operation of the normative environment with different e-contracting scenarios and enactments. Chapter 7 shows the feasibility of the answers to the second and third research questions.

- **Chapter 8** addresses the issue of norm enforcement by proposing an abstract model to represent the normative content of contracts, based on which an adaptive deterrence sanctioning model has been designed. Experiments have been made with a social simulation approach: the adaptation model was tested when facing different kinds of agent populations, where agents are parameterized regarding their risk tolerance and their social awareness when enacting the contracts they commit to. Chapter 8 provides a preliminary response to the fourth research question.

- Finally, **Chapter 9** looks back to the research contributions of this thesis, giving them a critical appreciation. It also identifies some lines of future research.

Part I

Background

Chapter 2

Electronic Contracting

Contracts are an essential element in business. A contract is a legally binding agreement between two or more parties. Typically, the parties entering in a contract commit to certain obligations in return for certain rights. In other words, a set of value exchanges is assumed to be taking place in contract enactment; in principle, every partner will benefit from such exchanges. According to contract law, contracts may be obtained by oral agreement (most purchases that we make every day are contracts of this form). We will, however, concentrate on discussing issues related with contracts that are obtained by explicit written agreements. These contracts may have an arbitrary complexity and give parties material evidence that prove the actual terms agreed upon at contract establishment.

Electronic contracting comprises all the activities related with running contractual relationships by electronic means. These means help overcoming the slowness and high costs of traditional paper-based contracts, and enable addressing emerging business paradigms based on dynamism and automation of trading relations. Angelov [6] has developed a thorough research on the business, legal and technological requirements for the development of highly automated e-contracting systems.

Legal aspects of e-contracting are already addressed in regulations such as the Directive on Electronic Commerce of the European Union [214] and its transposition to the Portuguese legislation [15]. In particular, such regulations admit the establishment of contracts between computers, that is, in an entirely automated way. But besides legal issues, the realization of e-contracting poses many technological and business challenges that have yet to be addressed.

Angelov identifies the benefits companies get from highly automated electronic contracting, and distinguishes different adoption levels of e-contracting processes, namely what he calls *shallow* and *deep* e-contracting [9]. The former exploits the use of information and communication technologies for carrying out, by electronic means, the same business processes that occur without these tools, and therefore maintaining the same level of involvement of people. On the contrary, deep e-contracting presupposes a higher level of automation, and leads to changes in the existing business processes. It also poses many research challenges concerning the supporting technologies. The carrying out of contracting activities through electronic means is still lacking supporting infrastructures that enable a higher level of automation.

In this chapter we dissect e-contracting into the several activities that may make part of this electronic approach to contracting. When doing so, we will provide an overview of the main research approaches devoted to such activities.

2.1 E-contracting Stages

Electronic contracting involves a set of activities that are centered on an essential concept: the electronic contract. Taking this into account, we identify three major stages within any e-contracting effort:

- *Pre-contractual*: This stage covers those activities related with information acquisition and identification of business opportunities. Electronic markets with matchmaking facilities are typical at this stage, before potential partners in a business contract start making preparations regarding such a relationship. The connection with the next stage might consist of an invitation to treat.

- *Contractual*: This stage comprises the negotiation of business terms and exchanged value provisions, which will be included in the contract. Market mechanisms for resource allocation, such as auctions and negotiation protocols, can and have been used to automate the achievement of deals with predefined contract templates. The negotiation of contractual terms is, however, a much more demanding activity to consider automating through electronic means. The output of this stage, in case of success, is a legally-binding agreement – a contract – which is the object of the whole contractual relationship. An electronic representation of the contract enables it to be automatically interpreted by computers.

- *Post-contractual*: After a contract has been established, it is time to proceed as agreed. Parties in the contract will enact the necessary activities to fulfill contractual terms. Contract monitoring tools may help on checking partners abidance to contractual terms, provided that the contract is represented electronically and in a way amenable to automatic processing. In case partners fail to comply with their obligations, the contract or the legislative system on which it is based may provide corrective measures to be applied.

Several authors have based their research on similarly defined contracting stages (e.g. [183][231]). In some cases the pre-contractual stage is further detailed into an information gathering and an intention phase [92][7]. In other cases the contractual phase is divided into the negotiation itself and the formal capture of negotiation outcomes in an e-contract amenable for automatic processing [178].

We will now provide a brief overview of some approaches addressing the main issues related with each of the three e-contracting stages as identified above.

2.2 Finding Business Partners

The pre-contractual stage is perhaps the most widely addressed by existing technological infrastructures. Since the beginning of e-commerce many on-line marketplaces have been developed, devoted to both domains of business-to-consumer and business-to-business. Business players have at their disposal the means to find potential business partners worldwide. Furthermore, there is already a high degree of automation at this stage. Matchmaking facilities already exist that automatically inform business players, based on their registered profiles, of new request-for-quotes (which comprise announced business opportunities).

Open marketplaces may be inadequate when the establishment of business relationships requires trust among individual business players. While most of these business platforms include some form of reputation mechanism that allow participants to provide feedback on the performance of contractual partners, this somehow does not seem to be enough in situations where partners really need to be acquainted with each other. Moreover, when one intends to automate further aspects of the e-contracting process, further supporting infrastructures are needed. This is important when companies want to address a highly dynamic market, seeking to react to business opportunities in a very prompt way, which is a perfect match for the requirement of a fast set-up phase for Virtual Enterprises [179][29].

In order to provide a higher degree of preparedness to address business opportunities, environments where groups of companies have prior knowledge and/or business experiences with each other have been proposed. These may consist of industry clusters of companies in the same or in complementary business domains. The concept of Virtual Enterprise Breeding Environment (VBE) [29][30] has been advanced as an association or pool of enterprises that are willing to cooperate (a keyword is *preparedness*) and that establish a long-term cooperation agreement. When a business opportunity arrives, a subset of the members of this pool may be selected, forming a VE. It is the role of the VBE to provide facilitating elements for boosting VE formation: a common infrastructure, common ontologies and mutual trust.

2.3 Negotiating E-contracts

Automated negotiation has been an active area of research for at least the last two decades, mainly in the multi-agent systems research community. Following the seminal works by Raiffa [181], and Rosenschein and Zlotkin [191], many researchers have devoted their efforts to devising negotiation protocols and strategies for automated negotiation among software agents.

Most research has, however, considered negotiation as a process of finding a mutually acceptable agreement on a multi-attribute search space. Every negotiable issue in a contract is therefore seen as an attribute in the space of possible deals, and in many cases research is kept at a very abstract level (e.g. [124]). Also, the automation of business negotiations using software agents has focused on negotiation as a partner selection process (e.g. [185]), not making a connection to the contract that formalizes the business agreement. But in business two negotiation stages are needed: the negotiation of exchanged values and the negotiation of their provisions, which is often called *contract drafting* [54]. Little work has been done on e-contract negotiation that takes into account contractual conditions such as deadlines, payment conditions, warranties, or sanctions. When making the connection with contract drafting, predefined contract templates are typically used.

The automation of negotiation using software agents is feasible only in relatively well-structured areas [231]. In most business settings negotiation will still need to be performed by humans in the foreseeable future. In these cases negotiation support systems [120] may have an important role to play.

The need for a starting ground in e-contract negotiations is acknowledged by several researchers (see, for instance, [183][196][231]). In fact, starting a negotiation where nothing is fixed represents a too ill-structured problem to consider automating. The importance of a *contract template* resides on its ability to provide a structure on which negotiation can be based. Certain kinds of business relations are formally typified and documented by relevant professional or legal bodies through model-form contracts. In this sense, instead of beginning from scratch a new contractual relation, prospective business partners can use an e-contract template, which can be seen as a contract outline containing domain-independent interaction schemata (business protocols) and variable elements to be filled-in with domain-specific data (such as exchanged values, prices, quantities, deadlines, and so on) resulting from a negotiation process (as in the approach suggested in [129]). If the negotiation is successful, an actual contract is produced that is an instantiation of the template. Templates thus provide a structure that allows negotiation, as a process of cooperative construction of a business relation, to be focused on those elements that are specific of the business at hand.

An interesting integration of both contract representation and negotiation is presented in [183], where auctions are automatically generated from declarative contract descriptions. The authors use a declarative language for representing both fully specified executable contracts and partially specified ones that are to be completed through negotiation. A contract template is composed of a fixed part – the so called *proto-contract* – and an identifiable set of issues to be negotiated; the template can then evolve through partial agreements, as each issue is successively fixed, until it becomes a fully-specified contract. The declarative description of partial contracts includes rules that describe the components of the contract left to be determined, expressing how they are to be negotiated. From this declarative setting, negotiation mechanisms are generated, which consist of AuctionBot auctions [237].

Other attempts to assist e-contract negotiation follow the negotiation support systems approach [120]. This seems to be more feasible in the short-term, while posing less demanding research challenges. In the collaborative networked organizations research community, supporting tools for virtual organization formation are being attempted [171], not having automation as a main concern.

Looking at the legal aspects of what may be included in a contract, and therefore what might be negotiated between business peers, we observe that contracting parties are free to create contracts with arbitrary content (this is expressed in article 405 of the Portuguese Civil Code [14] as *contractual freedom*). Therefore, although specific legislations may exist that regulate typical commercial relationships, such legislations are provided in order to facilitate parties when contracting, and usually not with the aim of constraining contractual content [118]. Parties are therefore allowed to contract around these “default rules” [48].

2.4 Representing E-contracts

Information and communication technologies have been used for representing e-contracts. However, current practice is largely devoted to giving natural language written contracts an electronic existence, which is supposed to facilitate searching and reading by human users. The possibility of having e-contracts that are automatically interpretable by machines has not been fully addressed by recent standardization efforts (e.g. LegalXML [157]).

The representation of law and regulations as computer programs has long been addressed by researchers (see [206] for a somewhat outdated survey). The possibility of reasoning about the application of legislations using artificial intelligence techniques has been actively pursued by many researchers. In particular, deontic logic [225] has played an important role in this research area [235][115]. The use of similar techniques to represent e-contracts (as studied by Daskalopulu [50]) is a natural extension of this work, since a contract can be seen as a regulation targeted at a specific group of individuals. One such attempt to represent deontic aspects of e-contracts is DocLog [213], which provides a semi-formal specification of contracts: an XML contract representation includes natural language clauses complemented with a structured normative description of such clauses. The aim of DocLog is to help human decision-makers in understanding the normative implications of certain business messages.

A number of research efforts have set higher standards in e-contract representation. More than having legal advice tools for legal practitioners, these efforts aim at automating the interpretation and execution of contractual content, enabling a higher level of business integration.

As mentioned before, deontic logic is the most widely used approach to model the normative content of contractual relations [197][194][212]. Deontic logic is a branch of modal logic also known as the logic of normative concepts. This logic embraces the notions of *obligation*, *permission* and *prohibition*, which are analogous to the modal concepts of necessity, possibility and impossibility, respectively. Although traditionally used to analyze normative reasoning in law, applications of deontic logic in computer science exist [235], not limited to the domain of legal knowledge representation and legal expert systems; other applications include, e.g., authorization mechanisms and electronic contracting.

Extensions to the original work on deontic logic (and particularly to what is known as Standard Deontic Logic [225]) have been made so as to allow its practical use. These include approaches to handle norm violations, such as the application of sanctions, also known as contrary-to-duties [32]: obligations that come into force in sub-ideal situations (i.e., when other obligations are violated). Other important extensions consider the use of conditional [222] and temporal [63][27] aspects: obligations are often made conditional on the occurrence of another event and have an associated deadline. Temporal logic [67] is often used to specify the semantics of deontic operators. Namely, Xu [238] uses propositional temporal logic for modeling monitorable contracts, and V. Dignum [66] proposes a Logic for Contract Representation based on branching-time temporal logic.

Using deontic logic to model the normative content of contracts typically consists of composing a set of normative statements including deontic operators (obligations, permissions or prohibitions), which are made conditional and have a temporal dimension (see e.g. the contract representation in the Electronic Contract Framework by Sallé [196], or the contract specification language used by Kollingbaum [129]). Another important issue in business contract obligations is directedness [194][212]: the identification of both the bearer (who is obliged) and the counterparty (to whom the bearer is obliged). Directed obligations are defined as commitments by Singh [230].

One possible representation for modeling obligations in contracts is based on the following structure: $\sigma \rightarrow O_{b,c}(\alpha \prec \delta)$. In this representation, when a state of affairs σ is the case, agent b is obliged towards agent c to bring about α before deadline δ. The characterization of the state of affairs may be such that obligations are raised on the fulfillment or violation of other obligations, making it possible to compose obligation chains that implement the desired contract enactment protocol. It also enables the specification of responses to contract violations (contrary-to-duties).

Other contract representation formalisms that include the handling of deontic operators have also been studied. Event Calculus [130][207] is a formalism for representing and reasoning about actions or events and their effects, consisting of a set of axioms in first-order predicate logic that defines the relationship between events, time-points and fluents. Event calculus has been used by Knottenbelt to define a contract language [126][125]. In his approach, events are defined as initiating or terminating fluents regarding the normative positions of agents within a contract. The evolution of obligations (their fulfillment or violation) are captured by rules specifying these event-fluent relationships.

An event-based representation of contracts is suggested by Abrahams [1]. Contracts (and business policies in general) are viewed as collections of occurrences that are stored in an occurrence store. An occurrence is an event that occurs at a particular point in time (or that occupies a period of time), and has participants acting in various roles (for instance, a purchasing occurrence has a purchaser, a purchased object and a seller). The occurrence store is an active database that checks newly added occurrences with existing policies, triggering contractually defined responses accordingly.

Processes and state machines have also been used to represent contracts. In Daskalopulu's approach [51], states represent the obligations that are in effect between the parties. Transitions correspond to the performance or non-performance of actions by parties. A similar approach is taken by Molina-Jimenez [164].

Some researchers try to build on rule-based markup languages to represent business rules. The SweetDeal system is introduced by Grosof [100] and described as a rule-based approach to the representation of business contracts (or more generally to represent e-commerce rules [98]). In this system, emerging semantic web standards for knowledge representation of rules (namely RuleML) are combined with ontology representation languages, taking into account business process description knowledge drawn from the MIT Process Handbook (a large repository widely used by industry business process designers). The system makes use of courteous logic programs [99]: sets of rules with explicit priority definitions among them, which are used for conflict handling.

2.5 Monitoring and Enforcing E-contracts

Once an e-contract is in place, parties are expected to follow their promises by fulfilling their part of the agreement. Software tools that support this stage of e-contracting have been proposed by several researchers. Such proposals range from simply monitoring the fulfillment of contracts, to providing facilities that allow agents to interact when enacting contracts, to developing mechanisms that make it possible to enforce parties to fulfill contracts.

In the real-world, when contractual obligations are assumed by parties they are typically not automatically enforced. For this reason, in non-electronic contracts contrary-to-duty definitions are not common. The violation of an obligation entitles the offended party to invoke legal power on a court of law, which may prescribe a secondary obligation to be imposed on the prevaricator [53]. Besides, parties are not willing to stipulate handling procedures for all possible circumstances, deferring them to when and if situations arise.

Although there are several research projects devoted to assisting and verifying e-business enactment, here we will only provide an overview of some of the approaches that can be found in the literature on e-contract monitoring and enforcement. In the next section we will discuss some of such projects. Automated monitoring of e-contracts requires a suitable representation of the contract. In the previous section we have identified some approaches available in the literature, and we here revisit some of them from a contract enactment perspective.

Xu [239] has developed an approach to monitoring multi-party contracts, focusing on formalizing a paper contract into representations suitable to monitoring purposes. For that, a paper contract is represented as a formal e-contract using temporal logic. The monitoring mechanism is designed to detect actual violations and to pro-actively detect and alert imminent violations. More specifically, the proposed contract model is composed of a trading process (including actions and commitments), a logic relationship specifying contract constraints (precedences among actions), and a commitment graph (a visual tool providing an overview of the commitments between partners). Pro-active monitoring [238] consists of checking and updating "guards of constraints", which capture the progression of constraint schemes and check what obligations remain to be realized. These guards are based on the pre-conditions and post-conditions of actions.

In his approach based on event calculus, Knottenbelt [127] provides a meta-interpreter that enables an agent to query what obligations are active with respect to a contract. Also, rules are included that capture the fulfillment or violation of obligations. Fluents pertaining to obligated facts are acknowledged in the system through a model of authorizations: each fluent is explicitly associated with the agents that are authorized to issue it. A non-authorized attempt will have no effect in the contract (that is, will not initiate any other fluent).

The involvement of a trusted third party in e-contract enactment is generally claimed. For instance, Kollingbaum [129] proposes a supervised interaction framework, where a trusted third party is included as part of any automated business transaction. Agents are organized in three-party relationships between two contracting individuals (a client and a supplier) and an authority that monitors the execution of contracts, verifying that errant behavior is either prevented or sanctioned. This authority enables the marketplace (where contract enactment takes place) to evaluate participants, keeping reputation records on the basis of past business transactions.

Sallé [196] proposes a *contract fulfillment protocol*, a collaborative protocol based on the lifecycle of normative statements. The idea is that, since contractual relationships are distributed, there is a need to synchronize the different views each agent has about the fulfillment of each contractual commitment. Each obligation has a set of states it might go through. An obligation can be refused or accepted by its bearer. After being accepted, the obligation may be canceled or complied with. These states are part of an obligation lifecycle model, and agents use this lifecycle to communicate their intentions on fulfilling contractual norms, allowing their contractual partners to know what to expect from them. This ability is referred to as *dynamic forecasting of partners' behavior*, and it enables a fluent and prompt execution of contracts, as agents do not have to wait for the fulfillment of their partners' obligations to start executing their own (hence the collaborative nature).

The general problem of coordinating business activities is addressed in [161], where Minsky proposes the utilization of a common law (a set of policies) to govern the interaction between agents representing different enterprises. The proposed *law governed interaction* is an interaction mode that guarantees the observance of a specified set of rules of engagement by each member of a group (such rules represent the law of the group, containing policies that reflect some prior contract between the group members). The law-enforcement mechanism is based on a set of trusted entities (controllers) that mediate the exchange of messages between the members of the group.

The real-world application of agents in automated contract fulfillment is challenged by the presence of complex legal issues and subjective judgments of agent compliance, as pointed out in a survey [109] exploring the use of agents in e-commerce. Still, some research on these matters has been made, for instance, by Daskalopulu [52], by proposing the presence of an e-market controller agent (a third party) to resolve disputes arising from subjective views on contract compliance, thereby playing the role of a judge. This agent holds a representation of the contract, and when a conflict occurs it collects evidence from the involved parties and obtains information from independent advisors, such as certification authorities, regulators, or controllers of other associated markets. Other authors have proposed subjective logic for addressing e-contract enforcement. Milosevic [159] highlights the distinction between discretionary and non-discretionary enforcement. While in the latter violations are dealt with indistinctively, the former is seen as a more realistic approach of reacting to a contract violation, whose extent may be variable and therefore demand for different corrective measures. The assessment of violations may be based on subjective logic as a means for combining parties' opinions.

2.6 E-contracting Endeavors

In this section we provide a brief overview of the most relevant undertakings on e-contracting that have been developed in the most recent years.

The **Elemental** project grew up from the work on e-contracting [158][93][159] carried out at the Distributed Systems Technology Centre of the University of Queensland. This work was one of the first addressing e-contracting with an integrative perspective, identifying e-contracting stages and activities such as negotiation, contract validation, monitoring and enforcement, together with a set of contract-related roles for B2B applications (e.g. contract repository, notary, contract monitor, notifier and enforcer). Focusing on automated contract management, a **Business Contract Architecture** has been designed where these components are implemented through Web Services. Although taking this general approach of identifying relevant services for e-contracting activities, the project seems to focus on bilateral relationships of the "extended enterprise" kind [131][160], wherein a dominant enterprise extends its boarders through business contracts that regulate the collaboration with secondary partners. This kind of inter-organizational collaborations require tight electronic links between organizations, which exchange messages that typically carry business documents. An XML-based Business Contract Language [165][131] was proposed aiming at describing contract semantics for the purpose of automating contract management activities. This language takes into account three categories of concepts: a *community* model defines the roles and their relationships, together with *policies* applied to those roles, and is used to define collaborative arrangements; *events* and *states* describe detailed behavior constraints and are the basis to support real-time monitoring; general language constructs enable the use of programming language constructs for more complex contract definitions.

SeCo (Secure Electronic Contracts) [193][92][96] started in 1998 at the University of St. Gallen and the University of Zurich, and was a project focused on merging business requirements on e-contracting with information technology security requirements and legal aspects. The project was initiated at the early stages of the legal acceptance of electronically created contracts and digital signatures, and as such puts an emphasis on developing a technical solution for secure and legally valid e-contracts. The phases of contracting were analyzed from a legal perspective, and a contracting framework for integrated business transactions was conceptualized including a set of contracting services, such as negotiation, monitoring and enforcement. The project had a strong emphasis on security aspects of e-contracting. Contracts are based on a so-called SeCo Container, used both for contract negotiation and for finalized contracts. A container includes a contract section (with the contract content and a digital signature block), a log section registering the events that occur during contracting (and that support the monitoring service) and a status section holding information about the current state of the SeCo Container.

COSMOS (Common Open Service Market for SMEs) [97] was a research project initiated in 1998 and lead by the University of Hamburg that aimed at developing a supporting platform for Internet business transactions based on a generic e-contracting system. Services to be facilitated to commercial partners include offer catalogues, brokerage, contract negotiation, signing and execution.

These services are meant to reduce transaction costs. A contract is defined in terms of four parts: the involved parties (who), the subject of the contract in terms of the transfer of items (what), the causal relationships defining when are which items to be delivered and which clauses to activate in case of non-compliance (how), and clauses addressing general terms and conditions as well as references to applicable regulations (legal). A contract template is initially defined including the "how" and "legal" clauses, together with roles, but without contract parties nor the exact obligations that are to be carried-on. Negotiation will fill-in the rest of the details and after the parties sign the contract it becomes executable and is transferred to a workflow execution engine.

CrossFlow (Cross-Organizational Workflow Support for Virtual Organizations) [128][113][112] started in 1998 with the University of Twente as a main partner, and aimed at cross-organizational workflow support in settings of service provider/consumer relationships. Organizations may delegate tasks in their workflows to external organizations (seen as service providers). In specific business sectors (vertical markets), standard types of services exist, opening the possibility for dynamic outsourcing based on a common understanding of those services. Standard services are parameterized for enabling flexibility in service enactment.

In CrossFlow, contracts are used as a means for flexible service outsourcing. A contract specifies the products or services to be exchanged and the rules of engagement. Contract negotiation was not a main issue in this project. The approach was based on standard form contracts in the context of specific markets. Contract templates are initially advertised by service providers via a trader, and searched by service consumers for contract matchmaking. Besides defining a business relationship, contracts are also used operationally in the execution phase. Process structures establish a mapping from service execution rules to workflow management systems in both service consumer and provider sides. This issue is representative of the emphasis of the CrossFlow project in contract enactment, focusing on the integration of cross-organizational workflow management systems. In this regard, it shares the same rationale as the **Coyote** project [49] developed at IBM Research Division, which tries to address business deals that are materialized into distributed long running applications spanning multiple autonomous business organizations. Both projects are concerned with the different processes that the companies perform during contract enactment. Coyote is also based on service-oriented technologies.

The CrossFlow project has eventually given rise to a follow-up enterprise at the Technical University of Eindhoven, where the work by Angelov [6] is most relevant. Angelov examined different business paradigms for the adoption of e-contracting [9], distinguishing between a shallower and a deeper approach, and discussing several implications of each. The investigations carried out tried to foster research on highly automated e-contracting systems, making a thorough study on the benefits of their development and adoption. The business, legal and technological requirements for this automation were also studied. A conceptual framework for B2B e-contracting support – the **4W framework** [8] – was developed that identifies and explores the concepts that are important for the design of an e-contracting system. A contract includes a set of participants (*Who*) exchanging obligations for rights (*What*) in a certain way (*hoW*); furthermore, it is established and enacted within a legal, geographical and business context (*Where*).

MeMo (Mediating and Monitoring Electronic Commerce) [56][199][180][231] started in 1999 with partners from The Netherlands, Spain and Germany. The project aimed at constructing a safe and trusted environment for promoting international e-commerce activities for SMEs. Tools for information discovery, protocol-based negotiations, and contract enactment monitoring, based on a business data repository, were included as parts of an electronic marketplace. The main focus was on supporting electronic negotiations among human negotiators, by analyzing the kinds of messages (e.g. offers, requests, quotations, counteroffers) that are exchanged in business scenarios, and the documents that these message exchanges deal with. The MeMo negotiation module enables human agents to structure their communication using a Formal Language for Business Communication [122], which is based on speech-act theory [17][202] and formal logic.

A few projects have concentrated on the legal aspects of e-contracting. The **eLegal** project [33] started in 2000 and its goals were to specify user requirements, implement legal support tools and promote enhanced business practices where electronic information exchange is contractually stipulated. The project had a strong focus in the construction industry. According to the project statement, the use of ICT in project-based business is lacking contractual frameworks that define legal conditions and contracts, and tools which provide such legal support. A set of contract negotiation tools has been developed that assists parties in establishing contracts that regulate data exchange rules and formats. **ALIVE** (Advanced Legal Issues on Virtual Enterprises) started in 2001 and was a research project aiming to identify and classify the legal issues arising from the emergence of the VE paradigm. Among them is the possible lack of legal personality of a VE and the implications this fact has on contract establishment with third parties. Furthermore, the project's final goal was to define a road-map for the resolution of such legal issues. An important deliverable of the project was the development of a Virtual Enterprise Legal Issue Taxonomy [200]. The close affinity between eLegal and ALIVE has lead to the organization of joint workshops.

CONTRACT [40][174][132][173][162] is a recent project on e-contracting that started in 2006. The aim of the project was to develop frameworks, components and tools which make it possible to model, build, verify and monitor distributed electronic business systems on the basis of dynamically generated, cross-organizational contracts. The project focused on specifying contracts for B2B interactions, on formal verification techniques for contracts, and on monitoring tools for contract-based systems. A theoretical framework for e-business applications based on binding contractual relationships was developed. A contracting language was proposed that tries to be less abstract than XML representations of human contracts (which cannot be used directly by computational systems for monitoring and control purposes, since they typically do not define how activities are to be monitored), and more general than service-level agreements specifications (which only allow for limited monitoring of computer-observable parameters or metrics). A Web Services framework has been designed implementing several contracting-related administrative roles for monitoring contract execution, such as observer, manager, contract storer or notary. Mechanisms for contract off-line verification, on-line monitoring and analysis have been designed that integrate with the developed frameworks.

Finally, an Europe-wide research initiative stemming from the multi-agent systems research community – **Agreement Technologies**[1] – has started in 2008 that aims at congregating efforts of researchers in the field to address issues related to a new paradigm for next generation distributed systems, based on the concept of *agreement* between computational agents. The issues tackled in this initiative are closely related to those of e-contracting, namely semantic alignment, norms, organizations, argumentation, negotiation, trust and reputation. All these issues are important aspects to take into account when aiming at automating contractual activities by employing software agents and multi-agent systems technology.

2.7 Conclusion

As shown in the previous section, the development of tools that support electronic contracting has gain much interest in the last decade. Looking back at the research projects that have dealt with some aspects of e-contracting, we may distinguish those that have made a more focused contribution from those that have tried to provide a more comprehensive perspective. Table 2.1 provides a summary of the aforementioned projects in this respect. We may observe that only two of the listed projects have a more comprehensive approach to e-contracting, namely the Business Contract Architecture (which is part of the Elemental project) and COSMOS. Even so, the former focuses on a particular collaboration setting (extended enterprise), and the latter reduces contract enactment to workflow activities. Almost every project that deals with contract monitoring assumes this simplification (CONTRACT, which is contemporary to the research reported in this thesis, seems to be an exception).

The investigations of Angelov [6], regarding the benefits of developing highly automated e-contracting systems, demand for further research on automating e-contracting activities. Although human involvement in the several stages that compose e-contracting is likely to be essential, there is still room for a number of developments that aim at automating specific e-contracting activities.

[1] `http://www.agreement-technologies.eu/`

Table 2.1: Focus of e-contracting projects

Project	*Main focus*
Business Contract Architecture	integrative perspective on e-contracting stages and activities; extended enterprise
SeCo	legal and security aspects of e-contracting
COSMOS	generic e-contracting system; semi-automated contract negotiation and enactment through workflows
CrossFlow	e-contract enactment in vertical markets through workflows
MeMo	e-negotiation support for human negotiators
eLegal	legal aspects for electronic information exchange; construction industry
ALIVE	legal aspects of Virtual Enterprises
CONTRACT	e-contract formal verification and monitoring tools

The efforts laid out by the MAS research community (namely through the Agreement Technologies initiative) aim at fostering research on making it possible to apply computational agents to several domains. Therefore, we see MAS as a promising approach to deliver automated tools that can be applied to the field of e-contracting. The research included in this thesis is a contribution towards that direction.

In the next chapter we provide an overview of MAS research with a bias on the use of norms and electronic institutions in agent systems. Our objective is to present the most relevant aspects of MAS research that are the cornerstones of the research reported in this thesis.

Chapter 3

Agents, Norms and Electronic Institutions

Since the conception of *agents and multi-agent systems* as a research field, researchers have been trying to come out with a list of properties that should characterize an agent as a intelligent entity. A widely accepted characterization [236] defines agents as entities that are capable of flexible autonomous behavior that exhibits *reactivity* (timely response to changes in the environment), *proactiveness* (goal-directed behavior) and *social ability* (interaction with other agents).

A main aspect of agenthood, which might go unnoticed in the previous paragraph, is *autonomy*: the ability of an agent to control its own behavior, i.e., to decide what to do in each situation that it is facing. This has been advocated as a distinguishing feature of agents when compared to object-oriented systems. One cannot simply "invoke a method" on an agent for it to do something. What one can do is to try to *influence its behavior*, e.g. by requesting the agent to do something. A whole theory of speech acts [17][202] has been applied in this field, giving rise to agent communication languages [75][81].

A richer perspective on autonomy is provided by Verhagen [224], who defines autonomy as an agent's level of independence with respect to another entity, be it the environment, other agents, or even the developers of the agent. Furthermore, Verhagen proposes an agent typology taking autonomy as the central issue:

- *Reactive agents* are autonomous only in the sense that they react without human intervention to the environment. Their model of deliberation is based on stimulus-response.
- *Plan autonomous agents* are able to choose, for a specific goal, how to achieve it. They have a repertoire of actions at their disposal that can be aggregated into plans that will achieve the objective.
- *Goal autonomous agents* are able to formulate, adopt and prioritize their

own goals. These agents are said to exhibit strategic reasoning, by comparing different states of the world in terms of degree of goal satisfaction.

- *Norm autonomous agents* chose which goals are legitimate to pursue based on a given system of norms. In a goal conflict situation, norm autonomous agents are able to change the norm system, and this ability is called normative reasoning (that is, reasoning about norms).

Being a multidisciplinary field as it is, multi-agent systems research has borrowed many concepts from different disciplines, such as economics, decision and game theory, philosophy, sociology or legal theory. Some inherited concepts are even addressed in more than one such areas. The cases of norm and normative behavior are some of such concepts. Before discussing the use of norms in multi-agent systems, we will start our discussion on norms by disambiguating the term *norm* as used in the social sciences and legal-theory.

3.1 Norms

In sociology, a norm is seen as a generalized expectation of behavior. The social sciences identify the concept of social structure as an underlying framework of norms that imposes a sense of order and predictability within a group of individuals. Typically, such norms are associated with roles that individuals play in a society. A role thus represents the way someone is expected to behave in particular situations, by having an associated set of normative expectations. Such roles do not have to correspond to well-defined social structures such as a formally specified organization. Rather, in society we may say that humans play roles every time they enter in a social relationship.

A clarifying view on social norms in social theory is Tuomela's [216], who distinguishes between *rules* (*r-norms*) and *proper social norms* (*s-norms*). The former are norms created by an authority and are subject to agreement. They can be either informal or formal, as can their associated sanctions. Proper social norms are based on mutual belief, and consist of conventions that may apply to a large group such as a whole society or to a specific group. These norms typically have implicit social sanctions attached to them.

In legal theory, norms are viewed as expressions of the obligations and rights that are connected to the role an individual has. This corresponds to the notion of *r-norms* as defined by Tuomela. Legal norms will also have formally specified sanctions. If we take Roman Law into account (which is the basis for the legal systems of many continental Europe countries), we also have well-established hierarchical normative systems that provide an explicit account for norms. These are typically organized through constitutions, laws and regulations.

Looking at the definition for norm in BusinessDictionary.com[1], we get precisely the two perspectives discussed above:

1. Informal guideline about what is considered normal (what is correct or incorrect) social behavior in a particular group or social unit. Norms form the basis of collective expectations that members of a community have from each other, and play a key part in social control and social order by exerting a pressure on the individual to conform. In short, “The way we do things around here.”

2. Formal rule or standard laid down by legal, religious, or social authority against which appropriateness (what is right or wrong) of an individual’s behavior is judged.

In the next section we will discuss the relatively recent introduction of norms in multi-agent systems research. We will then present an overview of some approaches for modeling norms.

3.1.1 Norms in multi-agent systems

In multi-agent systems research, social norms (sometimes referred to as conventions, social laws or joint commitments) have been used in the last two decades as a means to help improve coordination and cooperation [114][229][208][24][45]. As in real-world societies, norms provide us a way to achieve social order [34] by controlling the environment and making it more stable and predictable. Arguably, one of the main distinguishing factors among researchers using norms in multi-agent systems is the level of control one has over agents’ autonomy.

In fact, we may find in the literature two kinds of approaches to norms in multi-agent systems. Most initial research focused on norms as constraints on behavior [229][208], designed off-line and imposed in a top-down fashion. Therefore, in these settings norm abidance is not discretionary, i.e., it is not in the power of agents to decide whether to follow norms. The study of conventions to model multi-agent interactions follows a similar perspective. Conventions represent behavioral constraints [229][209][70] that intend to simplify an agent’s decision making process, by dictating courses of action to be followed in certain situations.

Another trend has then started that considered norm adoption as an important property of multi-agent systems [44][59][35][224][20]. The perspective is no longer to impose cooperation through normative constraints, but rather to study reasoning processes that lead agents to internalize norms at run-time and change their behavior accordingly.

Along this line, in [42] (also reviewed in [195]) three different views of norms as used in multi-agent systems research are identified:

[1] `http://www.businessdictionary.com/definition/norm.html`

- Norms as *constraints*: this view corresponds to the initial approaches, where the autonomy of agents is severely affected. In this case "norms" are always complied with, and can be seen as prohibitions (in the sense that constraints inhibit deviant behavior).

- Norms as *goals*: this view puts norms at the level of goals that an agent take into account in their reasoning cycle. It is up to the agent to choose among competing goals, and in this process the agent does not consider norms as more socially relevant goals. "Norms" may be abandoned whenever another goal becomes more relevant, and therefore there is no social control on the prevalence of norms over private goals.

- Norms as *obligations*: this view considers norms as first-order entities, a distinct element over which agents reason about. Furthermore, agents should be able to reason about the consequences of not fulfilling norms, which should cost more than abandoning ordinary goals.

We can say that unlike the assumptions present in the initial research on norms in multi-agent systems, agents are no longer assumed to be fully compliant with their norms. As noted by F. Dignum [59], norms should not be imposed, but should instead be used to influence agent decision-making, which allows us to keep agent autonomy: an agent might violate a norm in order to adhere to a private goal that it considers to be more important. This possibility of choosing to comply has been referred to as autonomous norm compliance in [150]. Some researchers (e.g. [59][26][4]) have expanded the well-known BDI agent architecture [182] to include normative reasoning.

More recently, *normative multi-agent systems* has risen as a research area on its own [23], and is at the intersection of two already established fields: *normative systems* and *multi-agent systems*. A normative system has been defined as a "set of interacting agents whose behaviour can usefully be regarded as governed by norms. Norms prescribe how agents ought to behave, and specify how they are permitted to behave and what their rights are" [116]. Also, "the norms allow for the possibility that actual behaviour may at times deviate from the ideal, i.e. that violations of obligations, or of agents' rights, may occur" [32].

Deontic logic is a formal tool to represent and reason about norms in a normative system, and is concerned with the normative notions of obligation, permission and prohibition. As noted by Jones and Sergot [115], deontic logic is particularly relevant for representing and reasoning about the distinction between ideal and actual behavior. Therefore, norm violations must be taken into account in any formalization of a genuine normative system. If agents were assumed to always conform to norms, the normative dimension ceases to be relevant. This view on the role of deontic logic makes it appropriate to model normative multi-agent systems.

Another line of research on norms in MAS, which is far from our intended purposes, is the emergence of collective social behaviors as an outcome of an ongoing interaction process among a group of individuals [123][209][204]. Norms are, in this sense, seen as conventions or patterns of behavior that emerge bottom-up from agent interactions. An attempt to provide an integrated view of norms can be found in [43].

3.1.2 Modeling norms

The conceptualization and specification of norms has been addressed by several researchers. In this section we present some of the most relevant contributions in this topic.

Following the research line that uses conventions to model multi-agent interactions, Esteva *et al.* [70] consider norms as procedural conventions, in the sense that they dictate what agents *must* do in such and such situations. These norms structure interaction protocols and limit the action repertoire of agents in pre-determined situations. More specifically, a set of so-called *normative rules* limit the action capabilities of agents when interacting within a *performative structure*, which is a transition graph composed of scenes and connections among scenes [2]. Some actions create commitments for future actions, and other actions may affect the paths an agent may take through the performative structure. A normative rule is activated by agent actions at specific scenes and prescribes obligations that hold. Obligations are here seen as actions that agents must do, and in order to enforce them the institution encapsulating the performative structure restricts the actions that an agent can perform while it has not fulfilled the obligations. This approach therefore does not account for true violation situations, in the sense that no sanctions are applied; instead, behavior restrictions will eventually lead agents to fulfill obligations.

In their work on modeling interactions in agent societies, V. Dignum *et al.* [66] look at such interactions as being structured in order to achieve some desired goals. For that, contracts are used to integrate a top-down specification of an organizational structure with the autonomy of participating agents, by making explicit the commitments regulating the enactment of roles by individual agents. A logical formalism based on deontic and branching-time logic is presented to specify social norms and interaction contracts between agents. A central issue in this proposal is norm violability. The authors argue that a Logic for Contract Representation must be able to reason about states in which an obligation has been violated, which is represented by an explicit *viol* predicate. In terms of deontic operators, their approach is based on the definition of conditional obligations with deadlines. Conditional obligations are obligations that only become active

[2] We will cover this approach more thoroughly when we address existing approaches to formalize electronic institutions.

if the precondition becomes valid. Deadlines identify situations when obligations may become violated. Sanctions (contrary-to-duties) are defined as obligations that hold in cases of violation of another obligation.

A categorization for norms in agent societies has been proposed by López-y-López and Luck [149][151]. They characterize norms by their *prescriptiveness* (norms tell an agent how to behave), *sociality* (norms apply in situations where more than one agent is involved) and *social pressure* (norms are subject to mechanisms that force agents to comply with them). Taking this into account, a *norm schema* is proposed including: *normative goals* (what ought to be done), *addressee agents* (directly responsible agents for the satisfaction of the normative goals), *beneficiaries* (agents that benefit from the satisfaction of normative goals), *context* (the particular circumstances where a norm applies), *exceptions* (situations in which addressees cannot be punished for not complying with norms), *rewards* (to be given when normative goals become satisfied) and *punishments* (to be applied when normative goals are not satisfied). The norm schema also includes some constraints on these components, saying that normative goals, addressees and context must all be non-empty sets, that the attributes describing both the context and exceptions must be disjoint to avoid inconsistencies and that the same should be the case for rewards and punishments. Based on this characterization, a classification for norms is proposed that considers three main categories:

- *Obligations and prohibitions* are norms whose purpose is to ensure coordination of individuals in a society. Agents adopt these norms once they become members of a society, not participating in their creation. The violation of these norms is always penalized (and therefore punishments must be non-empty).

- *Social commitments* are norms derived from agreements or negotiations between agents, which therefore participate on their creation (defining the normative goals, rewards and punishments). Social commitments come in pairs, denoting mutual relationships. Unlike obligations, social commitments are temporary, because they disappear once the normative goals are satisfied.

- *Social codes* are norms accepted as general principles and are complied with as ends in themselves, instead of being forced through punishments or rewards. An agent complies with social codes because of the perception other agents have of its social conformity.

The same researchers also introduce the notion of *interlocking norms*, which concerns the possibility of chaining norms: e.g. the compliance or non-compliance of a norm is a condition to trigger another norm. In other words, the context of the latter is dependent on the state of achievement of the normative goals of the former. Particularly relevant within interlocking norms are *enforcement* and *reward norms*. An enforcement norm is activated when another norm is unfulfilled and includes as normative goals the punishments associated with the unfulfilled norm. Similarly, a reward norm is activated when another norm is fulfilled and includes as normative goals the rewards associated with the fulfilled norm.

A different approach to the study of norms in organizations and institutions is that of F. Dignum [61] and Vázquez-Salceda [228]. These researchers attack the problem of norm verification by observing that norms, as used in real-world organizations, are often specified at a high level of abstraction. This characteristic is very common in legal texts, where vague terms are used to allow for proper interpretation within different contexts. Vázquez-Salceda and F. Dignum [228] argue that in order to check for norm compliance, abstract norms must be translated to a level where their impact on the institution can be described directly. They provide the illustrating example of a norm such as "*it is forbidden to discriminate on the basis of age*": it is necessary to bring this norm to the level of the concrete actions that agents may or may not execute. A multi-level framework for norms is proposed, from the most abstract level of the normative system to the final implementation of the organization – the Harmon*IA* framework [226]:

- The *abstract level* includes statutes of an organization, objectives, values and abstract norms.
- The *concrete level* includes the translation of abstract norms into concrete ones pertaining to actions described in terms of the organization's ontology, together with policies of an organization.
- The *rule level* is where concrete norms and policies are fully refined, linking the norms with the ways to ensure them; norms can be translated into restrictions on behavior or into triggers on unwanted behavior of the interacting agents.
- In the *procedure level* rules and policies are translated in a computationally efficient implementation to be used by agents.

Links are to be maintained between procedures and rules, and between rules and norms, in order to allow to track which abstract norms are related to which procedures.

Aldewereld [3] has followed this layered translation scheme to address the formalization of laws and regulations as abstract norms, which are then mapped to operational norms and implemented as institutional constraints. The translation of abstract norms into protocols, using *landmarks* at an intermediate level, is also proposed to address highly regulated systems, where protocols are to be used as guidelines to tell agents how certain tasks can be achieved with the lowest possible risk of norm violation. Landmarks define sets of propositions that ought to be true in certain intermediate states of a given protocol, allowing for a definition of that protocol in a two step process with an increasing amount of detail.

Vázquez-Salceda *et al.* have also looked into the implementation of norms and their enforcement. In [227] an operational semantics for the enforcement of norms is summarized in two possibilities: (i) defining constraints on unwanted behavior, or (ii) detecting violations and reacting to these violations. Therefore, a norm-aware environment can operate either preventively (making unwanted behavior impossible) or reactively. If we recall the issue of autonomy when discussing agent theory, and the associated lack of total control in a multi-agent system, we should see reaction to violations as the natural approach to take. In fact, Vázquez-Salceda *et al.* distinguish different kinds of addressees of norms, depending on the control the designer of the system has over interacting agents, and discuss different implementation guidelines for each situation. They also characterize norms (comprising obligations, permissions or prohibitions) by whether they refer to a state or an action, and by the fact that norms may be conditional or include a deadline, providing implementation guidelines for each situation. A *norm* specification is proposed that aggregates, together with what the authors call a *norm condition* (but which in fact includes both the conditions for norm activation and its prescriptions), violation conditions, sanctions and repairs. This makes up a heavy structure for each norm representation, making it harder to write norms in the first place; also, it causes the policies used for norm monitoring and enforcement to become spread among the norms themselves, which might be hard to maintain.

Also with an implementation concern, García-Camino *et al.* [87] provide a rule language for norms that is based on rules as used in production systems, further allowing constraints to be included in the state of the system. Basically, a rule has the form $LHS \rightsquigarrow RHS$, where *LHS* consists of atomic formulae (including constraints) that are to be matched with the current state of affairs, and *RHS* is composed of atomic formulae that are to be added or removed from the sate. Valid atomic formulae include agent actions (in this case utterances) and deontic operators (obligations, permissions, prohibitions) over utterances.

An approach to model and reason about normative positions based on Event Calculus [130][207] is proposed by Artikis *et al.* [13][12]. Besides obligations and permissions, this work considers also notions such as physical possibility and empowerments [117]. These normative positions are represented as fluents, and their interrelationships are expressed using event calculus predicates, which relate action occurrences with their effects in the values of fluents.

Cranefield [46][47] proposes a language called hyMITL±, a kind of temporal logic that combines Computational Tree Logic (CTL) with future and past-tense temporal operators and Metric Interval Temporal Logic (MITL). The language was designed to allow the expression of social rules with complex temporal properties, based on the $\downarrow$ binder operator from hybrid logics. A subset of this language allows social expectations to be expressed as rules that are conditional on observations, and that allow for run-time compliance monitoring. The language does not include deontic concepts such as obligations or permissions, addressing instead the broader notion of social expectations: constraints on the present and future that result from rules expressing social regularities, whether normative or not.

An important line of research regarding the representation of directed obligations is that of Singh *et al.* on commitments [210][37][36][230]. In this perspective, a commitment is a directed obligation from one agent to another for achieving or maintaining a state of affairs. A commitment has the generic form $C(x, y, p)$, where x is the *debtor*, y is the *creditor* and p is the *proposition* that x is committed to bring about. Furthermore, base-level commitments are distinguished from conditional commitments of the form $C(x, y, e \rightarrow p)$, where e is a *precondition* upon which the commitment becomes binding. This approach closely resembles the notion of conditional obligation as used by other researchers.

Following a similar commitment-based approach, Fornara and Colombetti propose a lifecycle of commitments in the context of an agent communication language [77][78]. The state of a commitment can evolve as an effect of a speech act, or of environmental events that change the truth value of the commitment's conditions or content. In this approach, norms are treated as event-driven rules that, when fired by appropriate events, manipulate (i.e., create or cancel) commitments affecting the agents that enact a specific role in an interaction.

3.2 Regulated Multi-Agent Systems

The regulation of environments where autonomous agents interact is an important research topic within the MAS research community. Different models have been proposed that try to give (artificial) agent societies some coordination and regulation infrastructures. In some approaches organizations are modeled as multi-agent systems that are assumed to cooperate so as to accomplish an overall goal. In this case a centralized design may be taken, producing a top-down specification of the roles agents may take and their associated norms. Other approaches take agents as representing independent self-interested entities, with no presupposed cooperation besides mere interaction efforts. These settings configure what are usually called *open* multi-agent systems [111], where heterogeneous agents arising from different sources behave and interact in ways that cannot be totally predicted in advance.

Before we discuss the notion of Electronic Institution as a means for delivering regulated multi-agent environments, in this section we identify some other approaches that are concerned with this issue.

Dellarocas [57] proposed the concept of **Contractual Agent Society (CAS)**, which is a metaphor for building open systems where unknown participants may enter and leave. The idea is that agents configure themselves automatically through a set of dynamically negotiated *social contracts*. These social contracts define the shared context for agent interactions, and should specify normative behaviors, exceptions and associated prevention and resolution mechanisms. The process of *socialization* is defined as the admittance of a new agent into the society, by negotiating the terms of a social contract defining the membership of this new agent. Sub-societies may be formed within the context of a CAS, requiring additional social contracts that inherit all policies of the CAS. Social contracts may also include negotiated social control mechanisms, identifying an authority party that will enforce such a mechanism. The metaphor of CAS is instantiated in a marketplace scenario.

Hannoun *et al.* [106] developed an organizational model for multi-agent systems called **MOISE**, which has been extended by Hübner *et al.* [107] into **MOISE$^+$**. These researchers view an organization as a global set of constraints which aims at conducting the agents' behavior according to what is socially intended. MOISE$^+$ is an organization-centered model where the structure and functioning of an organization are specified almost independently of each other, after which a deontic dimension links these aspects. The model is thus based on a three-folded specification:

- The *structural specification* defines agents' relations through the notion of roles, role relations and groups.
- The *functional specification* describes how a multi-agent system actually achieves its global goals, that is, how these goals are decomposed (in plans) and distributed to the agents (through missions).
- The *deontic specification* describes the roles' permissions and obligations towards the missions; it thus connects the social and functional parts.

In the deontic specification, a permission $per(\rho, m, tc)$ states that an agent playing role ρ is allowed to commit to mission m within a time constraint tc. An obligation $obl(\rho, m, tc)$ states that an agent playing role ρ ought to commit to m in the periods indicated by tc. An obligation entails a permission, and these deontic concepts are inherited throughout role hierarchies (from roles to role specializations).

V. Dignum [64] designed a model for agent societies called **OperA**. This model focuses on designed societies, with explicit objectives and structure, as opposed to emergent societies that result from "ad hoc" agent interactions, and tries to take into account both the structural and the dynamic aspects of an organization. The organizational framework has three components:

- The *organizational model*, defining the structure of the society based on roles and interactions.

- The *social model*, where agents adopt roles, in terms of agreements concerning their enactment.
- The *interaction model*, describing the possible interactions between agents.

Contracts are used to map the organizational model to the social model, and further to describe specific interaction agreements. Whereas the organizational model provides, through roles, the social, interaction, normative and communicative structures that shape the organizational characteristics of an agent society, the actual enactment of such roles by agents is fixed in *social contracts* that describe the capabilities and responsibilities of agents within the society. These social contracts compose the social model, which consists of an agent population composed of role-enacting agents. They negotiate interaction commitments that are fixed in *interaction contracts*. These contracts specify concrete interaction scenes, which comprise procedural instantiations (i.e. protocols) of scene scripts that are part of the interaction structures defined at the organizational model. Scene scripts provide a minimum set of requirements and constraints that allow an interaction to achieve its desired objectives. Agents choose, by negotiating interaction contracts, the best course of actions that achieves such objectives, which are specified as *landmarks* (states) that are partially ordered in *landmark patterns*.

In the next section we introduce and discuss the notion of electronic institutions, which can be seen as another approach towards having environments for regulated multi-agent systems. Given the importance of this topic in our research, we will try to provide an overview of the main approaches regarding the conceptualization of electronic institutions, but not before we seek to identify the roots of the concept.

3.3 Electronic Institutions

The concept of *institutions* has been studied from diverse fields such as law, economics, philosophy, or computer science. We will start by providing a brief overview of the most influential authors that have been followed when designing artificial or electronic institutions.

3.3.1 Institution: a multidisciplinary concept

In the economics field, North [167] defines *institutions* as “the rules of the game in a society”. Such rules are “humanly devised constraints that shape human interaction” and “define and limit the set of choices of individuals”. Institutional constraints include “formal written rules as well as typically unwritten codes of conduct that underlie and supplement formal rules”. These rules and informal codes may at times be violated – a central part of the functioning of institutions is the possibility of ascertaining violations and of reacting accordingly through punishments.

North also makes a distinction between *institutions* and *organizations*, which both provide a structure to human interaction. Organizations are taken to be "groups of individuals bound by some common purpose to achieve objectives". The institutional framework influences what kind of organizations come into existence and how they evolve. In turn, organizations influence how an institutional framework evolves. Therefore, there is a bidirectional interaction between institutions and organizations. The central point in North's research is precisely the effect institutions have on the performance of the economy. As he points out, the "major role of institutions in a society is to reduce uncertainty by establishing a stable (but not necessarily efficient) structure to human interaction". We may say that institutions evolve because of the changes humans introduce at the level of conventions, codes of conduct, norms of behavior, law and contracts between individuals.

In the philosophy field, Searle [202][203], looks at *institutions* as sets of *constitutive rules*. These rules are distinguished from *regulative rules*, which regulate antecedently existing activities. For instance, the rule stating that one should drive on the right-hand side of the road regulates driving, but driving as an activity can exist prior and independently of that rule (in fact, in some countries the rule does not even apply). On the contrary, constitutive rules create the very possibility of certain activities, besides regulating them. An example is the game of chess: the rules of chess make it possible to play the game, which does not exist prior to the rules. You are not playing chess unless you follow those rules. Moreover, Searle states that (systems of) constitutive rules have the form *"X counts as Y"*, or *"X counts as Y in context C"*. In this reading, a constitutive rule is said to assign a status-function Y to a fact X, possibly within a certain context C. Although it might seem otherwise, constitutive rules are *not* conventions, which according to Searle imply arbitrariness. Instead, a constitutive rule exists out of *collective intentionality*. Therefore, an institution refers to a set of concepts that exist thanks to a common agreement in a community of agents.

A further distinction that Searle makes is between *brute facts* and *institutional facts*. At the most basic level, brute facts pertain to physical reality, which exists independently of any human agreement on it. As an example, it is a brute fact that Mount Everest has snow near the summit. Differently, institutional facts rely on human institutions to exist. For instance, the fact that a piece of paper *counts as* a bill of five euros relies on the institution of money. Similar examples may be given for institutions like property, elections, marriages, and so on. Therefore, at a first level constitutive rules allow us to produce institutional facts from brute facts.

Furthermore, Searle notes that the structure of institutional facts is the structure of hierarchies of constitutive rules; in other words, the structure *"X counts as Y in C"* can be iterated. This means that status-functions can be imposed on other status-functions: the X term at a higher level can be a Y term from an earlier level. Here is a description that is pertinent for this thesis:

> "The Y term of one level can be the X term or the C term of the next level up or even of higher levels. Thus such and such an utterance as X1 counts as a promise Y1 in a context C1; but under certain circumstances C2, that very promise, Y1=X2, counts as a legally binding contract, Y2. Given the contract as a context, Y2=C3, a particular action as X3 can count as its breach, Y3. In the context of that breach, Y3=C4, a series of legal actions as X4 may count as a successful lawsuit, Y4, and hence have the function of remedying the breach or compensating for it." (from [203], p. 125)

Constitutive rules may therefore bring additional institutional facts from already established ones. The bottom line will nevertheless be a brute fact – there are no institutional facts without brute facts. Abstracting away from this understanding, we can also consider the set of institutional facts defined by an institution (considered as a set of constitutive rules) as a bottom line that can be used for the definition of other institutions. A notorious case, as Searle claims, is the fact that the institution of language is logically prior to other institutions – a primitive form of language is required to even express facts in other institutions.

In the computing and formal logic fields, Jones and Sergot [117] provide a formal account to *institutionalized power*, that is, to the normative notion of power as conferred by norm-governed institutions – "designated agents are *empowered* to create particular kinds of states of affairs by means of the performance of specified types of actions". Typically, such states of affairs are of special significance within that institution.

Theoretically, this model of institutionalized power is quite similar to the constitutive rules of Searle, although in this case the semantics of "counts-as" conditionals have an empowerment flavor: it is meant to distinguish valid from invalid actions, that is, those that produce a desired effect from those that do not because the enacting agent is not empowered by the institution. For example, in an auction house the auctioneer may utter the speech act "the item x is sold", which counts as a way of establishing that in fact item x is sold to a particular bidder. The auctioneer is said to have the power, within the auction house, to establish that item x is sold. The same action performed by an agent without this power has no effect.

The general form of a counts-as relation in the Jones and Sergot model is $E_x A \Rightarrow_s E_y F$, with the following reading: according to normative system or institution s, if agent x brings it about that A, then agent y sees to it that F. Agents y and x may at times be the same. Often, it would be appropriate to say that the agent y creating the state of affairs F is the institution or normative system itself. When x exercises his power the situation described by F becomes an *institutional fact* relative to institution s.

Following a long established observation from the legal field, a distinction is kept by Jones and Sergot between *institutionalized power*, *permission* and *practical possibility*. While institutionalized (or legal) power confers an institutional meaning to an agent's actions, this meaning will still apply regardless of the agent being permitted to act in such a way. It may be the case that the permission ceases to exist in specific circumstances, while the effects of action (which characterize the empowerment) may remain in place because the agent is still enacting a specific role. Of course that in such a case the agent may be sanctioned for doing something he was not permitted, but nevertheless the harm is done. Practical possibility refers to the physical ability needed to exercise legal power.

Jones and Sergot also generalize the counts-as relation, as used to define empowerment relations, to the notion of *institutional constraint*, denoted by statements of the form $D_s(A \rightarrow B)$. As argued, not all conditional sentences defined within a given institution are of the 'counts as' kind. Other conditionals describe relations of logical, causal or deontic consequence.

After recovering these insights on the notion of institutions, we now turn out attention to several approaches that deal with the issue of designing artificial or electronic institutions.

3.3.2 Agent-mediated institutions – IIIA

The research on electronic institutions at Institut d'Investigació en Intel·ligència Artificial (IIIA) started to take visibility with the work of Noriega [166], who proposed the building of computational environments as *agent-mediated institutions*. Such environments would allow heterogeneous agents to successfully interact by appropriately restricting their behavior. Furthermore, in order to guarantee that such restrictions were properly enforced, Noriega proposed the notion of *agent governors*, which are meant to ensure that external agents behave as supposed, i.e., adhere to the stipulated interaction protocols and rules. Agent-mediated institutions together with governors were claimed to allow a computationally realizable notion of accountability. While the work on agent-mediated institutions at IIIA has been primarily illustrated by means of the Fishmarket metaphor, it is meant as a more general approach to model any kind of *dialogical* institution – governed interaction environments where all agent interactions can be reduced to illocutions [202], and therefore protocols are described in terms of dialogues.

The formalization of the concept of agent-mediated institutions has been continued by the work of Rodríguez-Aguilar [188], and further on by Esteva [68], who mapped the specification of the concept to the construction of tools that allow for the development of electronic institutions. According to their perspective, an *electronic institution* (seen as an institutionalized electronic organization) is grounded on three main components [71][70]:

- *Dialogical framework*: In a computational realization of an institution, all interactions among agents are realized by means of speech acts (message exchanges). A dialogical framework thus specifies the illocutions that can be exchanged between the participants in an electronic institution, including an ontology definition for the concepts in a given domain. This framework includes also a specification of the (internal and external) roles that support the social interactions that may take place. Illocutions comprise a particle (the action), the sender and the receiver (in terms of agent/role pairs) and a content (an expression in some content language).

- *Performative structure*: Interactions between agents take place in agent group meetings – so-called *scenes* – that follow well-defined communication protocols. Each scene protocol is specified by a directed graph with nodes representing conversation states and directed arcs labeled with illocution schemes (illocutions that may contain variables) or timeouts. A performative structure captures relationships among scenes by establishing possible transitions, embodying the electronic institution with a more complex interaction structure. While a scene models a particular multi-agent dialogic activity, a performative structure enables the specification of more complex activities, e.g. by capturing causal dependencies among scenes, by defining synchronization or allowing parallelism mechanisms involving scenes.

- *Normative rules*: The performative structure constrains agent behavior both at intra-scene and inter-scene levels. An additional possibility is for agent actions within a scene to have consequences that either limit or expand its acting possibilities in subsequent scenes. Some actions create commitments for future actions, which are interpreted as obligations. Other actions may affect the paths an agent may take through the performative structure. These consequences are captured through normative rules. Obligations are represented as actions to be taken in specific scenes (that is, as arcs that an agent will be obliged to traverse within a scene).

It should be noted that the obligations prescribed by normative rules are not true normative constraints (following Castelfranchi's perspective [34]), in the sense that agents are not free to violate them. Instead, the electronic institution's performative structure and normative rules act as a constraining infrastructure that makes it impossible for agents to deviate from the desired behavior.

In terms of development tools, Esteva lead the efforts towards the development of the ISLANDER editor [69], a tool for the specification and verification of agent-mediated electronic institutions. ISLANDER facilitates the institution designer by combining graphical and textual specifications. It relies on a declarative textual language that allows for the specification of the aforementioned concepts. Its graphical user interface permits the edition of the graphical components of the scenes, performative structure and dialogical framework.

A further step in having a complete suite of applications for the development of agent-mediated electronic institutions was AMELI [72], a middleware that allows for the execution of electronic institutions as specified with ISLANDER. More specifically, AMELI provides an infrastructure that mediates agents' interactions while enforcing institutional rules. This is accomplished by interfacing every external agent with a governor (as suggested by Noriega), which filters out any unauthorized behavior (from the point of view of the performative structure and standing obligations).

The suite of developed applications for agent-mediated institutions [10] is arguably the most comprehensive one on the design, specification and development of electronic institutions.

More recently, García-Camino [84] worked on specifying norms for electronic institutions, where the prescribed obligations may have temporal restrictions and as such can be violated [85]. He also considered norms that may prescribe, besides obligations, also permissions and prohibitions over utterances, following a rule-based approach with constraints to represent norms and to apply their semantics [87]. García-Camino also proposed a rule language $\mathcal{I}$ [83] that allows to clarify the semantics of deontic modalities with respect to a computational system. The grammar for the language includes constructs for assigning different operational semantics to deontic operators, namely: *ignoring* events in order to avoid unwanted behavior, *preventing* conditions to hold in certain situations, *forcing* events or actions that are mandatory, or *expecting* events that an agent is obliged to generate but may even so violate. This different treatments of deontic operators allow to distinguish norms that can be violated from those that can not.

3.3.3 Institutions for electronic transactions – University of Utrecht

F. Dignum started to analyze institutions from the point of view of their importance in supporting electronic transactions [60]. Institutions are in this sense a means of enhancing the efficiency of transactions and trust among parties, e.g. by providing trusted information on potential partners, by defining and enforcing regulations on behavior during transactions, or by providing a legal backup for contracts that are closed within the institution.

A minimal infrastructure for electronic institutions should include banking facilities, communication facilities and an ontology. However, F. Dignum also suggests that the role played by an electronic institution is different according to the type of mechanism on which economic transactions are made: markets or hierarchies. In hierarchical relationships transactions tend to be direct between supplier and customer, and hence the role of an institution may be smaller; on the other hand, in markets the institution will typically be an intermediary between parties, possibly aggregating several functions such as brokering, setting interaction protocols (e.g. auctions), providing standard contracts, and so on.

A more detailed account of the roles that an institution may play within a coordination model (hierarchy, network or market) is given in [65]. An institution acts as a mediator who brings various skills and services, the most important of which is to regulate the interaction between members. But precisely because the way interaction happens between members depends on the coordination model, institutions need to be defined differently for each coordination model. It is the latter that determines the needed institutional roles, social norms and interaction forms in the society.

The main goal of a *market* model is to facilitate exchange between agents that provide services or resources, and typically there are several agents offering or needing the same type of resources. Facilitation roles that are identified for this setting are *registration* (of members), *machmaking* and *banking*. In a *hierarchy* model, some agents control a statically defined hierarchy, and work with predetermined suppliers instead of selecting them on a regular basis (the automotive industry is a paradigmatic example). Facilitating agents are dedicated to the overall control and optimization of the system's activities. Finally, in a *network* coordination model general patterns of interaction or contracts are used. Furthermore, coordination is achieved through the use of trusted third parties and according to well-defined rules and sanctions. Agents in a network society are said to enter a social contract in which they commit themselves to act according to such norms and rules. Facilitation agents monitor and register contracts, also helping other agents form those contracts. Roles such as *matchmaking*, *notary* and *monitoring* are relevant in this setting.

Also in collaboration with University of Utrecht, Vázquez-Salceda [228][226] makes a distinction between *institutions* and *organizations*. While institutions are seen as abstract entities that define sets of constraints, organizations are seen as instances of such abstract entities. An *institutional framework* thus exists within an organization of which different parties are members. A different term – *electronic organization* – is used to refer to a computational model that emulates a given organization by following its institutional framework.[3] As a definition, an electronic institution is seen as a set of templates that can be adapted, parameterized or instantiated to build an electronic organization. The Harmon*IA* framework [228] is precisely devoted to providing guidelines that assist this instantiation, by translating abstract norms to concrete norms and further to their realization as rules and procedures inside a specific organization.

[3]In this sense, the approach taken by the IIIA group would comprise electronic organizations, not institutions.

3.3.4 Institutional normative positions – Imperial College London

Research at Imperial College is based on the role of institutions on defining normative positions among (computational) agents. Stemming from the work by Sergot on normative positions [205] and institutionalized power [117], Artikis *et al.* [13][11] developed a framework where an institution is seen as an external entity ascribing institutional powers, normative positions and sanctions to agents.

According to Artikis, an open agent society may be viewed as a computational institution without goal orientation. In an open agent society members occupy roles and their behavior is governed by social laws. This society may include other groupings – institutions – that have their own constraints, roles, communication language, and so on. Taking into account different types of institutional constraints (as identified in [117]), the specification of such constraints is defined at three levels:

- *Valid actions*: what actions are to be treated as valid or successfully executed?
- *Permitted, obliged or prohibited actions*: what actions are legal?
- *Sanctions and enforcement policies*: how does the institution deal with illegal behavior?

The state of a particular institution contains information about the roles, institutional powers, normative positions and sanctions of its members.

Concerning implementation, action languages such as the Event Calculus [130][207] are used for representing the semantics of actions, in terms of what states of affairs they initiate or terminate. When capturing this semantics, the distinction between permission, physical capability and institutionalized power [117] is kept. Power relations (between an agent and an action) are used to capture valid actions in the context of an institutional state. Invalid actions are prevented of getting their intended effects on the institutional state. As for other constraints such as obligations and permissions, two possibilities are possible: either treat them as normative constraints, allowing agents to eventually violate them, or handle them as part of the specification of what counts as valid actions (something close to the original approach at IIIA). In the former case, sanctions can be imposed on violating agents.

3.3.5 Artificial institutions – Politecnico di Milano

Colombetti and colleagues investigate around the interplay between agent communication and the creation of institutional reality. Colombetti and Verdicchio [39] propose a distinction between *natural actions* (those that concern the activity of an agent in a physical environment) and *institutional actions* (which include performing speech acts). They define secondary actions as events intentionally brought about through the execution of another action, and relate this concept to the "counts as" relation of Searle. An institution defines contextual conditions for the application of instances of this "counts as" relation, and agents can exploit them to realize certain institutional actions.

Along this line, Fornara *et al.* [79] propose regarding an Agent Communication Language as a set of conventions to act on a fragment of institutional reality, defined in the context of an artificial institution. They model the context within which artificial agents operate as consisting of a set of entities that can have *natural* and *institutional attributes*. For instance, a book has a color (natural), and a price and owner (institutional). Institutional attributes can be affected by institutional actions. A model of authorizations (quite similar to Jones and Sergot's empowerments [117]) that are assigned to roles defines when institutional actions come into being.

An *(artificial) institution* [80] is here seen as an abstract description of *shared concepts and rules* that regulate a fragment of social reality, instead of the more usual definition of a framework that regulates the interaction among agents. An institution will be composed of a core ontology (institutional concepts and actions that operate on them), a set of authorizations (empowerments of agents to perform institutional actions), and a set of norms (imposing obligations and permissions on the agents).

3.3.6 Electronic institutions for B2B – LIACC

The concept of electronic institutions within this group, on which this thesis builds upon, started to gain form with the work of Rocha and Oliveira [185][187][186]. An *Electronic Institution* is seen as an agent-based software platform that enables automatic transactions among different participants according to a set of explicit rules and norms. These are meant to regulate the social behavior of participating agents. Additionally, the concept of a *Meta Institution* is introduced as a tool enabling the creation of appropriate electronic institutions, when specific needs in a business domain are detected.

The assistance provided by an electronic institution spans several activities that may comprise the realization of business transactions. Moreover, Rocha and Oliveira have devoted their attention to the phases that compose the lifecycle of Virtual Enterprises. An electronic institution is supposed to help in both the formation (setup) and operation phases of this business paradigm, by offering tools and services and by supervising business relationships created between parties.

3.4 Conclusion

The notion of Electronic Institution is a central one in this thesis. Although other research groups have built on this concept from different perspectives, they either lack implementation or coverage.

The IIIA approach to agent mediated institutions includes a complete suite of software that allows to build implementations of their notion of electronic institutions. However, the environments that can be created are completely predefined and leave almost no room for normative relationships created at run-time. The view of institutions as mechanisms to support electronic transactions, as approached at the University of Utrecht, lacks a computational implementation of the theoretically discussed concepts. The normative positions approach from Imperial College London, while providing an open perspective, lacks a supporting infrastructure assisting agents to obtain normative relationships. Finally, artificial institutions as viewed by the researchers at Politecnico di Milano are essentially focused on the study of the effect of agent actions (namely speech acts) on institutional reality.

In the rest of this thesis we will try to provide a detailed account to our approach towards the Electronic Institution concept. We see our approach as open-minded and practically guided. By open-minded we mean that it was our concern not to limit the applicability of the model, in terms of the normative relationships that potential users may be interested in establishing. By practically guided we mean that while providing a theoretical account to the concept, we are eager to apply our Electronic Institution vision to a real-world and increasingly relevant potential application area: e-contracting.

Part II

Research

Chapter 4

Electronic Institution for Agent-based B2B E-Contracting

In this chapter we provide an overview of the ingredients that we include in our own perspective of the Electronic Institution concept to be applied to the e-contracting domain.

As discussed in Chapter 2, e-contracting encompasses a set of activities that are carried out before, during and after an e-contract is established. These activities can be automated to a certain extent. If we want to effectively assist e-contracting from an integrative perspective, we must take into account activities that span a significant part of the process of managing contractual relations.

On the other hand, as discussed in Section 3.3.1, an *institution* is grounded on some notion of regulation, which is materialized through rules and norms. While some researchers take an abstract and immaterial perspective to institutions, we find it natural, when addressing electronic institutions, to follow a more proactive stance and ascribe to an EI the role of putting its regulations into practice. For that, two tasks are of paramount importance: monitoring and enforcement.

In order to embrace e-contracting activities in an EI, we look at the institutional regulations as governing the contractual relations that agents establish among them. Furthermore, we also endow the EI with a set of tools that assist computational agents in that very task of contract establishment. That is, besides monitoring and enforcing norms, institutional services will assist the coordination efforts among software agents which, representing different real-world entities, interact with the aim of establishing contractual relationships.

According to this perspective, the rationale of an EI is the following:

- To support agent interaction as a coordination framework, making the establishment of business agreements more efficient;
- To provide a level of trustworthiness by offering an enforceable normative environment.

An EI is thus a comprehensive framework that provides a set of institutional services, while assuring norm enforcement through a *normative environment*. The main role of this environment is to provide the necessary level of trust that enables heterogeneous and independently developed software agents to safely engage in business interactions. As the establishment of business agreements is central to our purpose, we consider an *evolving* normative environment, including the formalization of contracts through norms that the EI monitors and enforces. The EI will, nevertheless, provide a supportive normative framework (embedded into the normative environment), which agents can exploit in order to establish their contracts in a more straightforward fashion. This way, contracts can be under-specified by relying on this normative background.

The services that are to be provided by the EI are concerned with the activities identified as relevant for e-contracting (as discussed in Chapter 2). However, we put an emphasis on fostering research on the possible automation of those activities, following an agent-based approach. Furthermore, we identify a core infrastructure of the EI – the normative environment – which provides monitoring and enforcement services, and whose specification is the main focus of this thesis.

4.1 Services in an Electronic Institution

The services that we include in an EI cover a broad range of research issues within multi-agent systems research[1]. The research carried out in our group (LIACC-NIAD&R) has devoted attention to some of such services, namely:

- *Negotiation-mediation* [172][185]: This service aims at automating partner selection upon a business opportunity, and is based on appropriate negotiation protocols exhibiting important properties such as information privacy, qualitative feedback in proposal evaluation and adaptation.
- *Ontology-mapping* [155][154]: This service is crucial for the mutual understanding of software agents in open environments, consisting on establishing a semantic alignment between possibly disparate representations of similar business domain concepts.
- *Contract monitoring and enforcement*: This service addresses the post-contractual phase, by monitoring parties compliance with contractual terms and by applying specified sanctions in case of violations.
- *Computational Trust and Reputation* [219]: This service aims at aggregating trust information, namely building on past contractual behavior, which allows agents to make informed decisions regarding the selection of partners and/or the negotiation of contracts.

[1] In fact, they are accommodated into the Agreement Technologies research initiative, which started in 2008 (see Section 2.6).

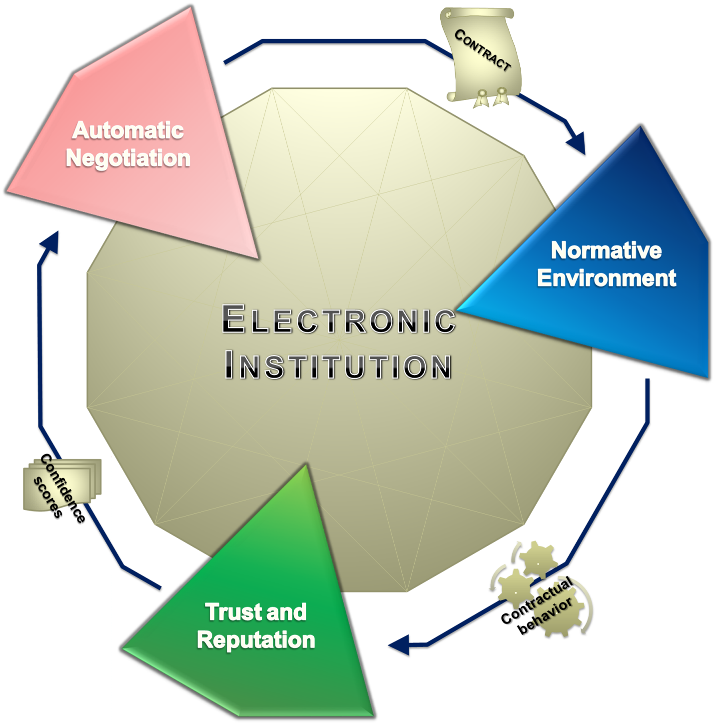


Figure 4.1: Electronic Institution Platform for B2B Contracting.

When designing an integrated approach that includes these services, we must also consider the interconnections among them. For instance, if ontology-mapping is to be useful, it must be aligned with the preliminaries of negotiation, allowing parties to resolve their semantic discrepancies before automated negotiation can take place [148]. Also, in order to move from the negotiation process to contract monitoring, contract drafting must be taken into account. In this case, we may instantiate contract templates (whose normative content may already be part of the institution's normative framework) with the outcome of negotiation, or include in the negotiation process itself the specification of contract clauses. Contracts resulting from successful negotiations may then be validated, registered and digitally signed, before being handed to a normative environment for monitoring and enforcement purposes. Finally, the way agents enact their contracts provides important information for trust building. A repository of trust and reputation information may then complete the circle by providing relevant inputs for future negotiations. The integration of all these stages (as depicted in Figure 4.1) has been addressed through the development of an Electronic Institution Platform for B2B Contracting.

In this thesis the issues of contract establishment, monitoring and enforcement are addressed.

4.2 Contractual Scenarios

Before going deep into the main subject of this thesis, which concerns the specification of an appropriate institutional normative environment for e-contracting, in this section we discuss the kinds of business relationships that we are looking at. This will provide some basis for understanding the rationale behind the research decisions and directions that we have taken, as described in subsequent chapters.

As expressed by Daskalopulu [54], electronic support is not likely to be needed in contractual situations of the simple purchase kind. In these situations checking compliance comes down to verifying whether goods are delivered on time and whether required payments are made. On the other hand, contracts that govern more lasting business relationships can benefit from tools that assist both contract formation and performance.

Furthermore, it is often the case that one contractual relation forms the business context for further contracts, as noted by Angelov [8]. These new contracts are dependent on the prior existence of that business relation, in the sense that contracts do not include every detail related to the (business) context within which they are formed. In other words, contracts are not always self-contained. An illustrating example (which can be found in [231]) is that of a wholesaler who negotiates with manufacturers about frame-contracts on a yearly basis; afterwards, the wholesaler will forward specific orders to the manufacturers within the boundaries of the agreed upon frame-contract.

This notion of contracts that span the lifetime of several subcontracts is observable in various business contexts, including the so-called Virtual Breeding Environments. In a VBE [29][30], a pool of enterprises establishes a base long-term cooperation agreement, which may be seen as a contract, depending on the courses of action that members commit themselves to (Romero *et al.* [189] study what kinds of governance rules can be defined for VBEs). A subset of VBE members will then be selected when a business opportunity arises, giving birth to a Virtual Enterprise that will be governed by a new negotiated contract. The VE governance rules may be inherited or instantiated from those governing the VBE within which the VE is created (the VBE fulfills in this way its preparedness rationale), but can also include other rules defined by VE partners [190].

A contract that governs a VE is also known as an *interchange agreement* [200]: a contract specifying the internal relations of a VE in terms of rules, liabilities and duties that exist between the VE members. In order to form a legally binding contract, the interchange agreement should be specific and unambiguous enough, instead of merely defining general goals or guidelines (the latter are termed framework clauses in [179]).

An important issue to take into account when attempting to automate contract monitoring is that parties' obligations should be expressed clearly, namely by referring to actions or states of affairs that can be verified through electronic means. It should be possible to determine whether parties' actions are in accordance with what was contractually stipulated.

While modeling supply chains using a multi-agent approach, Swaminathan *et al.* [211], as well as Verdicchio and Colombetti [223] identify three different kinds of flows between companies. *Material flows* from suppliers to clients concern the delivery of products. *Money flows* from clients to suppliers are meant for payments of products. Finally, *information flows* concern the exchange of several types of information in both directions, including documents such as orders, or supply updates such as expected delivery dates. These three kinds of elements are representative of the kinds of transactions that may occur in any B2B contract concerning the supply of products (in service contracts material flow is not evident, and quality-of-service metrics are typically employed [119][163]).

Summarizing, we are looking at contracts between a group of agents (representing business entities) that may be established in the context of standing contractual agreements. The commitments that bind agents together in such contracts regard exchanges of different kinds, which must take place in order for business to be enacted as planned.

4.3 Discussion

The research work presented in this chapter, namely the conceptual idea of an Electronic Institution as a service-providing infrastructure that includes a normative core, can be found in [136][134][133].

Our perspective on Electronic Institutions borrows from the University of Utrecht approach (see Section 3.3.3) the view of institutions as infrastructures providing services that are meant to enhance the efficiency of transactions. In respect to the main coordination models reported in [65], we assume a network model where the establishment of contracts is a central issue.

While we intend to obtain a computational realization of our concept (and have indeed progressed in that direction, within LIACC-NIAD&R, by building an Electronic Institution software platform), the cornerstone of our EI – its normative environment – has a significant distinction to that of other researchers as is the case of IIIA's approach (see Section 3.3.2). We want agents to be able to dynamically specify the normative relationships they are willing to establish, instead of providing a predefined normative scenario. Therefore, our perspective regards an EI not as an end *per se*, but as a means to facilitate both the creation and enactment of contracts between agents. The normative environment itself will be introduced, in the next chapter, with the concern of supporting contract formation, by incorporating and making available a background normative framework including norms that agents may choose to adopt or otherwise override.

Chapter 5

An Institutional Normative Environment

The use of an Electronic Institution as an infrastructure that enables regulation in multi-agent systems presupposes the existence of an environment including norms that somehow guide the way agents should behave. The purpose of this chapter is to present a formalization of an institutional normative environment addressing this need, envisioning its employment in e-contracting scenarios. We will start by providing an informal description of the environment we seek to develop.

5.1 Overview

The role of a normative environment is, besides providing a set of regulations under which agents' collective work is made possible, to check whether agents are willing to follow the norms they commit to (monitoring), and further to employ correction measures as a means of coercing agents to comply (enforcement). When building a computational realization of a normative environment, we must take into account the kinds of interactions that software agents will be able to carry out. If we are to build a self-contained environment where we are able to observe every agent activity, then what we get is a completely virtual world. On the contrary, we aim at providing an infrastructure that maps part of real world interactions (in an e-contracting environment) that take place between real entities. These entities are represented by software agents within the EI and, through them, are able to interact with the computational normative environment through speech acts.

We want to represent in the normative environment's structure the normative relations that correspond to contracts established by real world entities. This implies that instead of having a predefined normative structure, the shape of the environment will evolve and adapt to the actual contractual situations that are established. Furthermore, while monitoring the compliance to norms that apply to specific contracts, the normative environment will be recording a mapping from the relevant interactions (which concern e-contracting exchanges) that take place. The connection between real-world interactions and the institutional environment is made through illocutions (speech acts) that agents perform with the intent of informing the environment that certain contract-related events have occurred. With an appropriate interface between the normative environment and the statements that agents make, we incrementally build a state of *institutional reality*, which is an image of relevant real-world transactions that are through this means institutionally recognized.

The model we have developed is inspired by Searle's work on the construction of social reality [203] (as discussed in Section 3.3.1), where the use of constitutive rules allows facts to count as other institutional facts under some context. The former facts may be brute facts, or already established institutional facts. In our case, brute facts do not refer to physical reality, but rather to the illocutions agents perform towards the normative environment. These are our basic building blocks in order to build institutional reality. The main mechanism that we shall use in order to certify a real-world event by promoting it to institutional reality is that of empowerment as defined by Jones and Sergot [117] (also discussed in Section 3.3.1): informative illocutions performed by agents enacting specific roles count as certain institutional facts. In other words, agents enacting such roles are seen by the EI as trusted third parties, and are as such certified to obtain specific institutional facts. As an example, consider the acknowledgment of a monetary exchange: an agent enacting a banking role would be accepted as an entity asserting that such an exchange has in fact occurred, while the agent that was responsible to fulfill the transaction would not.

When providing a contract monitoring service, we take the stance that it is in the best interest of agents to publicize their abidance to contractual commitments. They do so by provoking the achievement of corresponding institutional facts. Depending on the constitutive rules that are in place, agents may need to interact with the institution's trusted third parties in order to convince the EI that they are in fact complying.

Since our normative environment is meant to handle several contracts at the same time, we need to provide some kind of structure to the corresponding normative relations inside the environment. Taking into account the contractual scenarios that we have delineated in Section 4.2, we build the environment's structure around a central notion: *context*. Each e-contract will be represented in the normative environment as a (normative) context. As a result, since institutional facts map exchanges that pertain to a certain contract, institutional facts are contextualized (i.e., inside the normative environment they are part of the respective context's state).

Because a contract can represent the business context for further contracts, we organize contexts hierarchically. Starting with a top institutional context within which every other context is formed, we allow contracts to inherit norms through this hierarchical structure. Furthermore, taking into account the aim of facilitating contract establishment, a supportive normative background may be included within this contextual structure. Predefined norms may therefore be inherited into sub-contexts. As in the contract law field, these norms work as "default rules" [48], and should be seen as facilitating rather than constraining contractual activity [118]. This means that such norms may be contracted around.

After providing an informal description of the structure of our institutional normative environment, we will now proceed to formalize it. We will start by providing a definition for the normative environment itself, together with our notion of context. A dissection of each component of the normative environment will follow. We will cover the specification of institutional reality and its handling, taking into account the aim of providing a contract monitoring and enforcement service. Afterwards, we will center our attention in the formalization of norms, explaining their semantics in a normative framework that includes context hierarchies supporting norm inheritance.

5.2 Normative Environment

We start by providing a definition for the normative environment as the aggregation of different kinds of rules, a normative state and norms.

5.2.1. Definition. *Normative Environment* $NE = \langle \mathbb{REA}, \mathbb{BF}, \mathbb{CR}, \mathbb{NS}, \mathbb{IR}, \mathbb{N} \rangle$
The normative environment NE of an EI is composed of a set $\mathbb{REA}$ of role-enacting agents, a set $\mathbb{BF}$ of brute facts, a set of $\mathbb{CR}$ of constitutive rules, a normative state $\mathbb{NS}$, a set $\mathbb{IR}$ of institutional rules that manipulate that normative state and a set $\mathbb{N}$ of norms, which can be seen as a special kind of rules.

Each of the elements that compose the normative environment will be defined below. Every agent participating in the EI will be enacting a role (see Def. 5.4.4), and will at times utter statements that are recorded as brute facts (see Def. 5.4.3). Constitutive rules (see Def. 5.4.5) process these brute facts in order to create, in the normative state (see Def. 5.4.2), institutional facts that represent an institutional recognition of a real-world event. While norms (see Def. 5.6.1) define the normative positions of each agent, the main purpose of institutional rules (see Def. 5.4.6) is to relate the normative state with the standing normative positions. A typical use of institutional rules is illustrated in Section 5.5, where they are employed to implement the semantics of deontic statements – rules monitor the normative state in order to detect the fulfillment or violation of deadline obligations. On the other hand, norms "produce" those deontic statements upon certain normative state conditions.

5.3 Contexts

Our model is based on structuring the normative environment with contexts. In this section we introduce the notion of context and context hierarchies.

5.3.1. Definition. *Context* $C = \langle PC, \mathbb{CA}, \mathbb{CI}, \mathbb{CN} \rangle$
A context C is a section of the normative environment representing the fact that a group of agents identified in set $\mathbb{CA}$ commits to a joint activity partially regulated by a set $\mathbb{CN} \subseteq \mathbb{N}$ of norms. A context includes a set $\mathbb{CI}$ of contextual info that makes up a kind of background knowledge for that context (see Def. 5.3.3). PC is the parent context within which context C is formed. Let $\mathbb{PCA}$ be the set of agents in context PC: we have that $\mathbb{CA} \subseteq \mathbb{PCA}$.

Contexts allow us to organize norms according to a hierarchical normative structure. Norm set $\mathbb{N}$ is partitioned into the several contexts that may exist, that is, sets $\mathbb{CN}$ for each context are mutually disjoint. Typically, we will have $\mathbb{CN} \subset \mathbb{N}$, in which case more than one context has a non-empty set $\mathbb{CN}$; only if all norms in $\mathbb{N}$ are defined in the same context we may have $\mathbb{CN} = \mathbb{N}$. A norm inheritance mechanism, as explained later (in Section 5.6.2), justifies the fact that the locally-defined set $\mathbb{CN}$ of norms only *partially* regulates the activity of agents in set $\mathbb{CA}$. We identify an *institutional* top level context from which all other contexts are (directly or indirectly) formed; every agent is associated with this institutional context.

We now introduce the notion of *sub-context*.

5.3.2. Definition. *Sub-context* $C' = \langle PC', \mathbb{CA}', \mathbb{CI}', \mathbb{CN}' \rangle$
A context C' is a sub-context of a context $C = \langle PC, \mathbb{CA}, \mathbb{CI}, \mathbb{CN} \rangle$, denoted $C' \lhd C$, if $PC' = C$ or if $PC' \lhd C$. When C' is either a sub-context of C or C itself, we write $C' \unlhd C$. From Def. 5.3.1 we also have that $\mathbb{CA}' \subseteq \mathbb{CA}$.

A sub-context defines a sub-activity committed to by a subset of the parent context's agents. Notice that the sub-context relationship is an explicit one. Every context is a sub-context of the *institutional* context.

We now turn to the definition of background information that may be defined as a foundational element of a context.

5.3.3. Definition. *Contextual info* $Info^C$
Contextual info $Info^C$ is a fully-grounded atomic formula in first-order logic comprising founding information regarding a context $C = \langle PC, \mathbb{CA}, \mathbb{CI}, \mathbb{CN} \rangle$. Looking at Definition 5.3.1, we have that $Info^C \in \mathbb{CI}$.

The $\mathbb{CI}$ set in a context definition is therefore composed of first-order logic atomic formulae that provide background information for that context.

The rationale of this context/sub-context relationship comes from the fact that in B2B a contract is often dependent on the existence of another business relation, which forms the business context for the new contract. Each contract contains a set of definitions regarding the role of the participants, the values to be exchanged (e.g. products and money) or any parameters defining their provision. In our model, these comprise information that is intrinsic and foundational to the context associated with this contract – hence the name *contextual info*.

5.4 Institutional Reality

Now that the central notion of context has been introduced, we can proceed to formalize the representation of institutional reality, as well as its manipulation.

5.4.1 Normative state

The normative state is organized through contexts, and composed of elements describing the current situation. We call every formula in $\mathbb{NS}$ an *institutional reality element*, or *IRE*. Each *IRE* pertains to a specific context within which it is relevant.

5.4.1. Definition. *Institutional reality element* IRE^C
An institutional reality element IRE^C is an occurrence regarding context C. We distinguish the following kinds of IRE^C with the following meanings:
$Ifact^C(f)$ – fact f is institutionally recognized as having occurred
$Time^C(t)$ – instant t has elapsed
$Obl_a^C(f \prec d)$ – agent a is obliged to bring about fact f until deadline d
$Fulf^C(obl)$ – obligation obl was fulfilled
$Viol^C(obl)$ – obligation obl was violated

The use of subscripts and superscripts in formulae is only a syntactical convenience – both contextual info and institutional reality elements are first-order atomic formulae (e.g., C could be used as a first argument in each of these formulae). While contextual info is confined to background information that is part of the context definition, institutional reality elements represent occurrences taking place after the context's creation, during its lifetime.

We consider institutional facts as *agent-originated*, since they are obtained (using constitutive rules) as a consequence of some agent action. The remaining elements are *environment events*, asserted in the process of norm application and monitoring. Our model of institutional reality is based on a discrete model of time. The *Time* elements are used to signal instants that are relevant to the context at hand. Obligations are deontic statements, and we admit both their fulfillment and violation[1]. Some of the *IRE*'s are interrelated: for instance, a

[1] In Chapter 6 we extend this to consider further states for obligations.

fulfillment connects an obligation to bring about a fact with its achievement as an institutional fact. These interrelations are captured with institutional rules.

5.4.2. Definition. *Normative State* $\mathbb{NS} = \{IRE_1^{C1}, IRE_2^{C2}, ..., IRE_n^{Cm}\}$
The normative state $\mathbb{NS}$ is a set of fully-grounded atomic formulae IRE_i^{Cj} in first-order logic.

The normative state will contain, at each moment, all elements that characterize the current state of affairs in every context. In that sense, $\mathbb{NS}$ could be seen as being partitioned among the several contexts that exist, as is the case with norms; however, *IRE*'s are not part of each context's definition, since they are obtained at a later stage, during the context's operation.

5.4.2 Constitutive rules

Constitutive rules make a connection between what is said and what is taken for granted. Before we formally define constitutive rules, we must provide a representation for informative illocutions, which comprise brute facts (elements of $\mathbb{BF}$, as in Def. 5.2.1).

5.4.3. Definition. *Brute fact* $Bfact(a, s)$
A brute fact $Bfact(a, s)$ is a statement s (regarding some context), uttered by agent a.

Since constitutive rules are, in part, based on the roles enacted by agents that perform illocutions, we also need to define a structure for storing role-enacting agents information (elements of $\mathbb{REA}$, as in Def. 5.2.1).

5.4.4. Definition. *Role-enacting agent* $Rea(a, r)$
An agent a enacting role r is represented as $Rea(a, r)$.

We are now in a position to define constitutive rules.

5.4.5. Definition. *Constitutive rule* $CR ::= FactConds \rightarrow Ifact^C$
A constitutive rule *CR* specifies, for one or more brute facts and the roles of the agents creating them, or from one or more institutional facts, what institutional fact should be added to the normative state. The constitutive rule's conditional is a conjunction of patterns of *Bfact* and *Rea*, or a conjunction of patters of $Ifact^C$, which may contain variables:

$$\begin{array}{rcl} FactConds & ::= & BruteFactConds \mid InstFactConds \\ BruteFactConds & ::= & Term \mid Term \wedge BruteFactConds \\ Term & ::= & Bfact(...) \mid Rea(...) \\ InstFactConds & ::= & Ifact^C(...) \mid Ifact^C(...) \wedge InstFactConds \end{array}$$

The constitutive rule's conclusion is an institutional fact that is allowed to contain bounded variables.

When using brute facts, the context ascribed to the obtained institutional fact is extracted from brute fact statements, which are expressed according to some well-defined ontology. Constitutive rules must inspect brute facts in order to obtain the corresponding institutional facts. This corresponds to mapping between two different ontologies: one for brute facts and another for institutional facts. Although agents are free to utter any statements they like, only a subset of these will have a meaning inside the EI, namely those that are recognized by constitutive rules.

When the brute fact patterns are matched against the brute facts in $\mathbb{BF}$ using a first-order logic substitution Θ, the fully-grounded institutional fact obtained by applying Θ to the rule's conclusion is added to the normative state. The same is true when matching institutional fact patterns against the normative state $\mathbb{NS}$.

Examples of constitutive rules

Since we are concerned with business scenarios involving transactions, we define three main roles that provide a connection to real-world events: a *bank* role certifies money exchanges (i.e., payments); a *delivery tracker* role certifies material exchanges (i.e., product deliveries); a *messenger* role certifies information exchanges (e.g. documents). The following comprise examples of constitutive rules regarding these kinds of transactions (variables are represented by lowercase identifiers).

Consider a situation in which an agent ought to make a certain payment to another. Although the agent may claim to have paid its debt, that does not make it the case. However, if a trusted financial third party agent states that a currency transfer referring to a certain context (e.g. a sales contract) has taken place, it would be safe to consider that the payment took place:

$$\begin{array}{l} \mathit{Bfact}(b, \mathit{CurrencyTransf}(\mathit{Context}{:}\,\mathit{ctx}, \mathit{Ref}{:}\,r, \mathit{From}{:}\,\mathit{a1}, \mathit{To}{:}\,\mathit{a2}, \mathit{Amount}{:}\,\mathit{am}))\wedge \\ \mathit{Rea}(b, \mathit{Bank}) \\ \rightarrow \mathit{Ifact}^{\mathit{ctx}}(\mathit{Payment}(\mathit{Ref}{:}\,r, \mathit{From}{:}\,\mathit{a1}, \mathit{To}{:}\,\mathit{a2}, \mathit{Amount}{:}\,\mathit{am})) \end{array}$$

In this example, $\mathit{CurrencyTransf}(\mathit{Context}{:}\,\mathit{ctx}, ...)$ is a statement that a currency transfer of *am* from *a1* to *a2* has taken place regarding overall transaction r within context *ctx*. The purpose of using a reference for the overall transaction is that of enabling the occurrence of similar brute or institutional facts within the same context (the reference, e.g. an invoice number, assigns a unique identifier to each payment, in this case).

We can also say that if both agents (the payer and the receiver) state that a payment took place, it would also be safe to conclude the associated institutional fact:

$$\begin{array}{l} \mathit{BFact}(\mathit{a1}, \mathit{Paid}(\mathit{Context}{:}\,\mathit{ctx}, \mathit{Ref}{:}\,r, \mathit{To} : \mathit{a2}, \mathit{Amount}{:}\,\mathit{am}))\wedge \\ \mathit{BFact}(\mathit{a2}, \mathit{Collected}(\mathit{Context}{:}\,\mathit{ctx}, \mathit{Ref}{:}\,r, \mathit{From}{:}\,\mathit{a1}, \mathit{Amount}{:}\,\mathit{am})) \\ \rightarrow \mathit{Ifact}^{\mathit{ctx}}(\mathit{Payment}(\mathit{Ref}{:}\,r, \mathit{From}{:}\,\mathit{a1}, \mathit{To}{:}\,\mathit{a2}, \mathit{Amount}{:}\,\mathit{am})) \end{array}$$

In this case, $a1$ is stating that, within context ctx and regarding overall transaction r, he has paid am to $a2$, while $a2$ is stating that he has collected am from $a1$. It might be questionable whether the receiver's statement would be sufficient.

The same approach may be defined concerning material exchanges:

$$
\begin{aligned}
&Bfact(dt, Delivered(Context\!:ctx, Ref\!:r, From\!:a1, To\!:a2, Product\!:p, Quantity\!:q)) \wedge \\
&Rea(dt, DeliveryTracker) \\
&\rightarrow Ifact^{ctx}(Delivery(Ref\!:r, From\!:a1, To\!:a2, Product\!:p, Quantity\!:q))
\end{aligned}
$$

$$
\begin{aligned}
&BFact(a1, Sent(Context\!:ctx, Ref\!:r, To\!:a2, Product\!:p, Quantity\!:q)) \wedge \\
&BFact(a2, Received(Context\!:ctx, Ref\!:r, From\!:a1, Product\!:p, Quantity\!:q)) \\
&\rightarrow Ifact^{ctx}(Delivery(Ref\!:r, From\!:a1, To\!:a2, Product\!:p, Quantity\!:q))
\end{aligned}
$$

In this case, $a1$ is stating that, within context ctx and regarding overall transaction r, he has sent q units of i to $a2$, while $a2$ is stating that he has received q units of i from $a1$.

Finally, a similar approach for information exchanges:

$$
\begin{aligned}
&Bfact(m, MsgDelivered(Context\!:ctx, Ref\!:r, From\!:a1, To\!:a2, Msg\!:msg)) \wedge \\
&Rea(m, Messenger) \\
&\rightarrow Ifact^{ctx}(MsgDelivery(Ref\!:r, From\!:a1, To\!:a2, Msg\!:msg))
\end{aligned}
$$

$$
\begin{aligned}
&BFact(a1, MsgSent(Context\!:ctx, Ref\!:r, To\!:a2, Msg\!:msg)) \wedge \\
&BFact(a2, MsgReceived(Context\!:ctx, Ref\!:r, From\!:a1, Msg\!:msg)) \\
&\rightarrow Ifact^{ctx}(MsgDelivery(Ref\!:r, From\!:a1, To\!:a2, Msg\!:msg))
\end{aligned}
$$

In this case, $a1$ is stating that, within context ctx and regarding overall transaction r, he has sent message msg to $a2$, while $a2$ is stating that he has received message msg from $a1$.

In principle, any information exchange could be treated with this approach. However, since constitutive rules can be based on institutional facts, we can acknowledge business-related documents such as orders, invoices and so on. As an example, we may acknowledge the placement of an order using the following constitutive rule:

$$
\begin{aligned}
&Ifact^{ctx}(MsgDelivery(Ref\!:r, From\!:a1, To\!:a2, Msg\!:msg)) \wedge \\
&msg = Order(Product\!:p, Quantity\!:q) \\
&\rightarrow Ifact^{ctx}(Order(Ref\!:r, From\!:a1, To\!:a2, Product\!:p, Quantity\!:q))
\end{aligned}
$$

The advantage of having this constitutive rule is that it makes it easier to express norms that are based on orders: instead or relying on a "message" fact, we can make an explicit reference to an order document. What is important to retain from this last example is that by inspecting the contents of a message (which may have a more or less complex structure) we can infer specific documents to have been delivered.

5.4.3 Institutional rules

Given the "contextualization" of the normative state, we are now able to introduce the notion of institutional rules. These rules allow us to update the normative state of the system, by capturing interrelations between IRE^C's.

5.4.6. Definition. *Institutional rule IR ::= Antecedent → Consequent*
An institutional rule *IR* defines, for a given set of conditions, what other elements should be added to the normative state. The rule's *Antecedent* is a conjunction of patterns of IRE^C (see Def. 5.4.1), which may contain variables; restrictions may be imposed on such variables through relational conditions over expressions. We also allow the use of negation (as failure):

$$\begin{array}{rcl} \mathit{Antecedent} & ::= & \mathit{Term} \mid \mathit{Term} \wedge \mathit{Antecedent} \mid \neg \mathit{Antecedent} \\ \mathit{Term} & ::= & \mathit{IRE}^C \mid \mathit{relational\text{-}condition} \end{array}$$

The rule's *Consequent* is a conjunction of IRE^C which are not deontic statements (referred to as $IRE\text{-}^C$), and which are allowed to contain bounded variables or expressions using bounded variables:

$$\mathit{Consequent} \quad ::= \quad \mathit{IRE}\text{-}^C \mid \mathit{IRE}\text{-}^C \wedge \mathit{Consequent}$$

When the antecedent matches the normative state using a first-order logic substitution Θ, and if all the relational conditions over variables hold, the atomic formulae obtained by applying Θ to the consequent of the rule are added to the normative state as fully-grounded elements.

5.5 Monitoring Deadline Obligations

One of the most important roles for institutional rules is in monitoring obligations that agents acquire through norms. As defined in Def. 5.4.1, obligations that are part of the normative state are deadline obligations, in the sense discussed in [62].

In this section we specify, in a first attempt, the semantics of deadline obligations and further implement this semantics through institutional rules. This approach will be revised in Chapter 6, where we take a different perspective on modeling contractual obligations.

5.5.1 Semantics of deadline obligations

In order to model the semantics of deadline obligations, we will make use of linear temporal logic (LTL) [67] with a discrete time model: time has an initial moment with no predecessors, and is infinite into the future. Let $x = (s_0, s_1, s_2, ...)$ be a timeline, defined as a sequence of states s_i. The syntax $x \models p$ reads that p is true in timeline x. We write x^k to denote state s_k of x, and $x^k \models p$ to mean that p is true at state x^k.

We will use a weak version of the *before* LTL operator B, where the left operand is mandatory, while the right operand is not:

- $x \models (p\ B\ q)$ *iff* $\exists_j(x^j \models p \land \forall_{k<j}(x^k \models \neg q))$.

This before operator is not strict, since it also succeeds when both operands become true at the same time point. We will also make use of the henceforth LTL operator G:

- $x \models G\ q$ *iff* $\forall_j(x^j \models q)$

The following relationships express how deadline obligations are fulfilled or violated:

$$Obl_a^C(f \prec d) \land (Ifact^C(f)\ B\ Time^C(d)) \Rightarrow G\ Fulf^C(Obl_a^C(f \prec d)) \tag{5.1}$$

$$Obl_a^C(f \prec d) \land (Time^C(d)\ B\ Ifact^C(f)) \Rightarrow G\ Viol^C(Obl_a^C(f \prec d)) \tag{5.2}$$

With this approach, we are basically depending on which comes first: the deadline or the accomplishment of the obliged fact (this is in line with the semantics followed by Broersen *et al.* [27]).

We want obligations not to persist after the deadline. This allows us to model both cases of legal obligations, namely obligations that stand even when violated and those that do not. For instance, Dignum *et al.* [62] provide the following examples: an obligation to pay for a fine will persist if it is not fulfilled until the deadline, while an obligation to submit a conference paper will not persist after the submission deadline (because submitting makes no sense at that stage). For modeling a standing obligation, the obligation can be reinstated after a violation is detected, by means of an appropriate sanctioning norm.

This property can be stated as follows: a fulfilled obligation cannot be violated anymore, and a violated obligation cannot be fulfilled anymore.

$$Obl_a^C(f \prec d) \land Fulf^C(Obl_a^C(f \prec d)) \Rightarrow G\neg Viol^C(Obl_a^C(f \prec d)) \tag{5.3}$$

$$Obl_a^C(f \prec d) \land Viol^C(Obl_a^C(f \prec d)) \Rightarrow G\neg Fulf^C(Obl_a^C(f \prec d)) \tag{5.4}$$

These relationships remove the obligation's effect after it has been fulfilled or violated.

5.5.2 Implementation with institutional rules

Institutional rules, being used to maintain the normative state of the system, allow us to implement the semantics of deadline obligations, as defined above. The $Fulf^C$ and $Viol^C$ terms in Def. 5.4.1 are meant to allow us to reason about the fulfillment and violation of obligations as soon as they occur, by defining norms that take these elements into account (see Def. 5.6.1). Institutional rules enable the specification of conditions for fulfillment and violation detection.

According to the deadline obligation semantics described above, namely (5.1) and (5.2), we may have the following institutional rules (variables are represented by lowercase identifiers):

$$Obl_a^c(f \prec d) \wedge Ifact^c(f) \wedge \neg Time^c(d) \rightarrow Fulf^c(Obl_a^c(f \prec d)) \tag{5.5}$$

$$Obl_a^c(f \prec d) \wedge Time^c(d) \wedge \neg Ifact^c(f) \rightarrow Viol^c(Obl_a^c(f \prec d)) \tag{5.6}$$

A problem with these rules is that it is unclear what is supposed to happen when both the institutional fact and the deadline temporal reference hold. That is, in a situation where the institutional fact occurs exactly at the deadline, we should have both elements $Ifact^c(f)$ and $Time^c(d)$ added to the normative state before rules are evaluated again – in this case, none of the rules will apply. If we add a rule for dealing specifically with this case, we may say:

$$Obl_a^c(f \prec d) \wedge Ifact^c(f) \wedge Time^c(d) \rightarrow Fulf^c(Obl_a^c(f \prec d)) \tag{5.7}$$

However, this is not acceptable, because the rule will still apply when the deadline is obtained before the institutional fact. Instead, we need to keep the property that after being violated, the obligation cannot be fulfilled anymore (as in (5.4) above). We could thus replace rule (5.5) for:

$$Obl_a^c(f \prec d) \wedge Ifact^c(f) \wedge \neg Viol^c(Obl_a^c(f \prec d)) \rightarrow Fulf^c(Obl_a^c(f \prec d)) \tag{5.8}$$

It is tempting to also explicitly state that violations can only occur if no fulfillment was achieved before. We could write:

$$Obl_a^c(f \prec d) \wedge Time^c(d) \wedge \neg Fulf^c(Obl_a^c(f \prec d)) \rightarrow Viol^c(Obl_a^c(f \prec d)) \tag{5.9}$$

However, when taken together with (5.8), this would imply that a simultaneous occurrence of $Ifact^c(f)$ and $Time^c(d)$ could bring either a fulfillment or a violation! We therefore must join (5.6) with (5.8). (Notice that the pairing of (5.5) with (5.9) would bring a violation in the simultaneity case.)

This approach is based on the implicit assumption that rules are evaluated at every normative state update, otherwise we may get unwanted results. For instance, let us assume that the following are elements of the current normative state, and that they are obtained in this order: $Obl_A^C(F \prec D)$, $Time^C(D)$ and

$Ifact^C(F)$. In this case, we have reached the deadline before the institutional fact was added. However, if rules are applied only at a time later than D, the violation would go unnoticed: rule (5.8) would apply, while rule (5.6) would not.

If we are to relax the rule evaluation policy, the two rules for fulfillment and violation detection must become independent. A way to achieve this is to enrich *IRE*'s that are used by monitoring rules, by time-stamping them: $Ifact^C(f)^t$, $Fulf^C(obl)^t$ and $Viol^C(obl)^t$ now include an explicit temporal reference t for their occurrence. We are now in a position to rewrite our monitoring rules as follows:

$$Obl_a^c(f \prec d) \wedge Ifact^c(f)^t \wedge t < d \rightarrow Fulf^c(Obl_a^c(f \prec d))^t \tag{5.10}$$

$$Obl_a^c(f \prec d) \wedge Time^c(d) \wedge \neg(Ifact^c(f)^t \wedge t < d) \rightarrow Viol^c(Obl_a^c(f \prec d))^d \tag{5.11}$$

The shortcoming of this approach is that it is directly applicable only when considering temporal deadlines (in previous rules the $Time^c$ element could be easily replaced by, e.g., an institutional fact used as a deadline).

Another assumption that we make regards the instantaneous recognition of each *IRE*. That is, an institutional fact occurring at time T is added at that same instant T to the normative state. Were that not the case, we could get into situations where certain violations would need to be retracted as new knowledge is acquired, otherwise inconsistencies might occur (an extra $\neg Time^c(d)$ test in rule (5.10) would avoid getting an obligation both violated and fulfilled, although the normative state would still be inconsistent). The approach that we take in Chapter 6 will allow us to make this model less demanding in terms of fact recognition.

5.6 Normative Framework

As introduced in Section 5.1, and formalized in Section 5.3, contexts provide us an hierarchical structure through which norms may be inherited, from contexts to sub-contexts. This structure enables us to assert a contract in the context of another previous contractual agreement, and will allow the new contract to inherit applicable norms from that previously established relationship. Looking at this feature from a broader perspective, the EI may include, in its institutional (top level) context, a supportive normative framework embedded in the normative environment. This will facilitate contract establishment by providing predefined norms that regulate specific types of contracts. However, agents should be able to contract around these regulations, which should be seen as "default rules" [48]. In order to enable this approach, in this section we properly formalize the use of norms throughout our context hierarchies, including how and when they are to be inherited.

5.6.1 Norms

Before explaining the mechanism for applying norms in our hierarchical context structure, we must start by defining the very concept of *norm*. As with institutional reality elements, norms are contextual.

5.6.1. DEFINITION. *Norm* $N^C ::= Situation^{C'} \rightarrow Prescription^{C'}$
A norm N^C is a rule with a deontic consequent, defined in a specific context C. The norm is applicable to a context $C' \trianglelefteq C$. The norm's situation is a conjunction of patterns of $Info^{C'}$ and $IRE^{C'}$. Both kinds of patterns are allowed to contain variables; restrictions may be imposed on such variables through relational conditions. We also allow the use of negation (as failure):

$$\begin{array}{rcl} Situation^{C'} & ::= & Term \mid Term \wedge Situation^{C'} \mid \neg Situation^{C'} \\ Term & ::= & Info^{C'} \mid IRE^{C'} \mid relational\text{-}condition \end{array}$$

The norm's prescription is a (possibly empty) conjunction of deontic statements (obligations), which are allowed to contain bounded variables or expressions using bounded variables; the same context C' is assigned to prescribed deontic statements:

$$\begin{array}{rcl} Prescription^{C'} & ::= & \epsilon \mid Obls \\ Obls & ::= & Obl^{C'}(...) \mid Obl^{C'}(...) \wedge Obls \end{array}$$

Conceptually, the norm's *Situation* can be seen as being based on two sets of elements: *background (Sb)* and *contingent (Sc)*. Background elements are those that exist at context creation (founding contextual info), while contingent elements are those that are added to the normative state at a later stage. This distinction will be helpful when describing the norm inheritance model.

We emphasize the distinction between the context where the norm is defined, and the context to which the norm applies. While, in order to make the model simpler to follow, we defined a norm as being applicable to a specific context, in Section 5.6.3 we will enlarge the norm's scope, which will in part clarify the usefulness of the model.

5.6.2 Contextual norm inheritance and defeasibility model

In this section we define the semantics of norms and formalize a model for norm inheritance and defeasibility within the sphere of a supportive normative framework. We start by exploring norm applicability according to the normative state. For that, we make use of the notion of *substitution* in first-order logic. We write $f \cdot \Theta$ to represent the result of applying substitution Θ to atomic formula f.

5.6.2. DEFINITION. *Norm activation*
A norm $N^C = S^{C'} \rightarrow P^{C'}$, applicable to a context $C' = \langle PC', \mathbb{CA}', \mathbb{CI}', \mathbb{CN}' \rangle$, is said to be *activated* if there is a substitution Θ such that:

- $\forall_{c \in Sc}\ c \cdot \mathit{\Theta} \in \mathbb{NS}$, where Sc is the set of contingent conjuncts ($IRE^{C'}$ patterns) in $S^{C'}$; and
- $\forall_{b \in Sb}\ b \cdot \mathit{\Theta} \in \mathbb{CI}'$, where Sb is the set of background conjuncts ($Info^{C'}$ patterns) in $S^{C'}$; and
- all the relational conditions over variables hold.

We are now able to define the notion of conflicting norm activations, as follows.

5.6.3. Definition. *Norm activation conflict*
Let Act_1 be the activation of norm $N_1^{C1} = S_1 \to P_1$, obtained with substitution Θ_1, and Act_2 the activation of norm $N_2^{C2} = S_2 \to P_2$, obtained with substitution Θ_2. Let $NS_1 = \{c \cdot \mathit{\Theta}_1 | c \in Sc_1\}$, and $NS_2 = \{c \cdot \mathit{\Theta}_2 | c \in Sc_2\}$, where Sc_1 and Sc_2 are the sets of contingent conjuncts of S_1 and S_2, respectively. Both NS_1 and NS_2 represent fractions of the whole normative state $\mathbb{NS}$. Norm activations Act_1 and Act_2 are in *conflict*, written $Act_1 \bigotimes Act_2$, if $NS_1 = NS_2$ and either $C1 \lhd C2$ or $C2 \lhd C1$.

Succinctly, we say there is a norm activation conflict if we have two applicable norms activated with the same fraction of the normative state and defined in different contexts. Notice that the fact that both norms are activated with the same contextual *IRE*'s already dictates that the norm contexts, if different, have a sub-context relationship (there is no multiple inheritance mechanism in our normative structure). This becomes clearer when taking into account the sub-context (Def. 5.3.2) and norm (Def. 5.6.1) definitions: a context has a single parent context, and a norm N^C applies to a context $C' \unlhd C$.

In principle, all norm activations are defeasible, according to the following definition.

5.6.4. Definition. *Norm activation defeasance*
A norm activation Act_1 for norm N_1^{C1} *defeats* a norm activation Act_2 for norm N_2^{C2} if $Act_1 \bigotimes Act_2$ and $C1 \lhd C2$.

A defeated norm activation is discarded, that is, the defeated activation is not applied to the normative state fraction used for activating the norm. Only undefeated norm activations will be effective: the substitution that activated the norm is applied to its prescription and the resulting fully-grounded deontic statements are added to the normative state (recall that there are no free variables in the prescription part of norms). At this point we should stress the fact that we do not talk about norm defeasance, but rather *norm activation* defeasance. Thus, the defeasance relationship may only materialize on actual norm applicability.

5.6.3 Norm contextual target

Let us restate the purpose of having defeasibility in norms as presented in the model above: to have a normative background that can fill-in details of sub-contexts, which are created later and can benefit from this setup by being under-specified. Thus, part of the normative environment's norms will be predefined, in the sense that they are preexistent to the applicable contexts themselves. What we need is to typify contexts in order to be able to say that a norm applies to a certain *type* of contexts, instead of a particular individual context. This way, a norm might be defined at a super-context and applicable to *a range of* sub-contexts (of a certain type) to be subsequently created.

We can do this adaptation by considering the context identifier C, as used in the patterns composing norms, as a pair $id : type$, where id is a context identifier and $type$ is a predefined context type. In a norm $N^C = S \rightarrow P$ (see Def. 5.6.1), patterns of $Info^{C'}$ and $IRE^{C'}$ inside S, as well as obligations inside P, will be rewritten to accommodate this kind of context reference, optionally using a variable in place of the context id. For instance, an $IRE^{id:T}$ pattern, where id is a variable, would match IRE's of any sub-context of type T. When activating a norm with this kind of pattern, the substitution Θ (as used in Def. 5.6.2) would have to bind id to a specific context identifier; every further occurrence of id within the norm becomes thus a bounded-variable.

This approach allows us to maintain our definitions of norm activation conflict and defeasance, with minor syntactical changes. Typifying contexts also opens up the possibility to define context-type hierarchies (much the same way as we have class hierarchies in object-oriented programming): we may define a general context type from which every other context (directly or indirectly) extends. Using this approach, we can specify general policy norms that span multiple context types; for instance, we may define a general norm saying that in principle every deadline violation is supposed to bring some kind of penalty to the violator, regardless of the context type at hand. This norm will then be defeasible inside particular contexts, if one wishes to do so.

5.6.4 Example

The exploitation of our normative environment, especially when considering its role in providing a supportive normative background, demands for knowledge engineering efforts focused in building appropriate normative structures that may be of use in recurrent real-world contracting situations. When doing so, it will be necessary to specify, for each type of contract, an appropriate ontology for contextual information (see Def. 5.3.3) needed when creating a contract of that type. Although the normative core of a contract may be inherited, the specific information that instantiates a particular contract must be expressed in a per-contract basis (in many cases this information can be regarded as a set of contract parameters). Users must therefore know this "interface" when instantiating a contract of a predefined type.

While in Chapter 7 we will illustrate the use of the normative environment in some stereotyped situations, in this section we sketch a simple example that will help on understanding how norm inheritance and defeasibility work in practice. For that, we will make use of the timestamped version of institutional reality elements (as introduced in Section 5.5.2), namely $Ifact^{id:C}(f)^t$ and $Fulf^{id:C}(obl)^t$.

The scenario that we use here is that of a "supply agreement": each of a group of companies (agents) provides different resources that may need to be combined in order to present a value-added offering to third-parties. For that, they will agree on a standing agreement for the supply of their products to any contract partner according to specific conditions. The contextual information that is to be provided when forming a contract of this predefined type (a supply-agreement *SA*) has the following format:

$$SupplyInfo^{id:SA}(Agent : a, Product : p, UnitPrice : up)$$

This formula indicates that agent a commits to supply product p at unit price up.

This type of contract is regulated by the following norm, defined in the top institutional context:

$$\boxed{N_1^{Inst}}$$
$$Ifact^{id:SA}(Order(Ref : r, From : a1, To : a2, Product : p, Quantity : q))^t \wedge$$
$$SupplyInfo^{id:SA}(Agent : a2, Product : p, UnitPrice : up)$$
$$\rightarrow$$
$$Obl_{a2}^{id:SA}(Delivery(Ref : r, From : a2, To : a1, Product : p, Quantity : q) \prec t + 2) \wedge$$
$$Obl_{a1}^{id:SA}(Payment(Ref : r, From : a1, To : a2, Amount : q * up) \prec t + 2)$$

Norm N_1^{Inst} states that in any supply-agreement, when an order is placed that matches the supply information of the receiver, the latter is obliged to deliver the requested products, while the ordering agent is obliged to make the corresponding payment; both exchanges are due at the same deadline (within 2 days).

Now, let *SA3* be an actual supply agreement between three agents: *Jim*, *Sam* and *Tom*. This will translate into a context *SA3*:*SA* in the normative environment, with $SA3{:}SA \lhd Inst$. Suppose we have the following founding contextual information for context *SA3*:*SA*:

$$SupplyInfo^{SA3:SA}(Agent : Jim, Product : P1, UnitPrice : 1)$$
$$SupplyInfo^{SA3:SA}(Agent : Sam, Product : P2, UnitPrice : 1)$$
$$SupplyInfo^{SA3:SA}(Agent : Tom, Product : P3, UnitPrice : 1)$$

Furthermore, the agents also chose to add a few norms to their contractual relationship, as follows.

$\boxed{N_{1}^{SA3:SA}}$

$$\begin{aligned}&\mathit{Ifact}^{SA3:SA}(\mathit{Order}(\mathit{Ref}\!:r, \mathit{From}\!:a1, \mathit{To}\!:\mathit{Jim}, \mathit{Product}\!:p, \mathit{Quantity}\!:q))^{t} \wedge\\&\mathit{SupplyInfo}^{SA3:SA}(\mathit{Agent}\!:\mathit{Jim}, \mathit{Product}\!:p, \mathit{UnitPrice}\!:up) \wedge q > 99\\&\rightarrow\\&\mathit{Obl}_{\mathit{Jim}}^{SA3:SA}(\mathit{Delivery}(\mathit{Ref}\!:r, \mathit{From}\!:\mathit{Jim}, \mathit{To}\!:a1, \mathit{Product}\!:p, \mathit{Quantity}\!:q) \prec t+5) \wedge\\&\mathit{Obl}_{a1}^{SA3:SA}(\mathit{Payment}(\mathit{Ref}\!:r, \mathit{From}\!:a1, \mathit{To}\!:\mathit{Jim}, \mathit{Amount}\!:q * up) \prec t+2)\end{aligned}$$

Norm $N_{1}^{SA3:SA}$ expresses the fact that agent *Jim*, when receiving orders with more than 99 units, has an extended delivery deadline.

$\boxed{N_{2}^{SA3:SA}}$

$$\begin{aligned}&\mathit{Ifact}^{SA3:SA}(\mathit{Order}(\mathit{Ref}\!:r, \mathit{From}\!:\mathit{Sam}, \mathit{To}\!:a2, \mathit{Product}\!:p, \mathit{Quantity}\!:q))^{t} \wedge\\&\mathit{SupplyInfo}^{SA3:SA}(\mathit{Agent}\!:a2, \mathit{Product}\!:p, \mathit{UnitPrice}\!:\_)\\&\rightarrow\\&\mathit{Obl}_{a2}^{SA3:SA}(\mathit{Delivery}(\mathit{Ref}\!:r, \mathit{From}\!:a2, \mathit{To}\!:\mathit{Sam}, \mathit{Product}\!:p, \mathit{Quantity}\!:q) \prec t+2)\end{aligned}$$

$\boxed{N_{3}^{SA3:SA}}$

$$\begin{aligned}&\mathit{Fulf}^{SA3:SA}(\mathit{Obl}_{a2}^{SA3:SA}(\mathit{Delivery}(\mathit{Ref}\!:r, \mathit{From}\!:a2, \mathit{To}\!:\mathit{Sam}, \mathit{Product}\!:p, \mathit{Quantity}\!:q) \prec d))^{t} \wedge\\&\mathit{SupplyInfo}^{SA3:SA}(\mathit{Agent}\!:a2, \mathit{Product}\!:p, \mathit{UnitPrice}\!:up)\\&\rightarrow\\&\mathit{Obl}_{\mathit{Sam}}^{SA3:SA}(\mathit{Payment}(\mathit{Ref}\!:r, \mathit{From}\!:\mathit{Sam}, \mathit{To}\!:a2, \mathit{Amount}\!:q * up) \prec t+2)\end{aligned}$$

These two norms express the higher position of agent *Sam* who, as opposed to other agents, is only obliged to pay after receiving the merchandise.

Table 5.1 shows what might happen in different normative states. Lines labeled with $\bigotimes$ in the first column show which norm activation conflicts come about (and how they are resolved) when the institutional reality elements of their previous line (labeled with $\mathbb{NS}$) are present. Lines labeled with $\mathbb{NS}'$ show the normative state after applying the defeating norm activation.

The first example shows a trivial situation which falls into the general institutional norm. Therefore, in this case $\mathbb{NS}'$ contains $\mathbb{NS}$ together with the prescription of norm N_{1}^{Inst} (after applying the substitution that activated the norm). In the second example there is no conflict since norm $N_{1}^{SA3:SA}$ is not activated because of a variable restriction (the ordered quantity is within regular limits). The third example shows the particular case of an order that is handled by the defeating norm $N_{1}^{SA3:SA}$. The fourth and fifth examples are in sequel, and illustrate *Sam*'s advantage in being obliged to pay only after the delivery has been fulfilled – first $N_{2}^{SA3:SA}$ obtains *Tom*'s obligation to deliver and then $N_{3}^{SA3:SA}$ prescribes *Sam*'s obligation to pay.

In each case we rely on refraction (a principle used in rule-based systems) to avoid firing a defeating norm more than once on the same activation (which would otherwise happen since our normative state is monotonic).

Table 5.1: Different normative states and norm activation conflicts.

$\mathbb{NS}$	$Ifact^{SA3:SA}(Order(Ref:1, From:Jim, To:Tom, Product:P3, Quantity:5))^{1}$
$\otimes$	none, N_1^{Inst} applies
$\mathbb{NS}'$	$Ifact^{SA3:SA}(Order(Ref:1, From:Jim, To:Tom, Product:P3, Quantity:5))^{1}$
	$Obl_{Tom}^{SA3:SA}(Delivery(Ref:1, From:Tom, To:Jim, Product:P3, Quantity:5) \prec 3)$
	$Obl_{Jim}^{SA3:SA}(Payment(Ref:1, From:Jim, To:Tom, Amount:5) \prec 3)$
$\mathbb{NS}$	$Ifact^{SA3:SA}(Order(Ref:2, From:Tom, To:Jim, Product:P1, Quantity:5))^{1}$
$\otimes$	none, N_1^{Inst} applies
$\mathbb{NS}'$	$Ifact^{SA3:SA}(Order(Ref:2, From:Tom, To:Jim, Product:P1, Quantity:5))^{1}$
	$Obl_{Jim}^{SA3:SA}(Delivery(Ref:2, From:Jim, To:Tom, Product:P1, Quantity:5) \prec 3)$
	$Obl_{Tom}^{SA3:SA}(Payment(Ref:2, From:Tom, To:Jim, Amount:5) \prec 3)$
$\mathbb{NS}$	$Ifact^{SA3:SA}(Order(Ref:3, From:Tom, To:Jim, Product:P1, Quantity:100))^{1}$
$\otimes$	$N_1^{SA3:SA}$ defeats N_1^{Inst}
$\mathbb{NS}'$	$Ifact^{SA3:SA}(Order(Ref:3, From:Tom, To:Jim, Product:P1, Quantity:100))^{1}$
	$Obl_{Jim}^{SA3:SA}(Delivery(Ref:3, From:Jim, To:Tom, Product:P1, Quantity:100) \prec 6)$
	$Obl_{Tom}^{SA3:SA}(Payment(Ref:3, From:Tom, To:Jim, Amount:100) \prec 3)$
$\mathbb{NS}$	$Ifact^{SA3:SA}(Order(Ref:4, From:Sam, To:Tom, Product:P3, Quantity:5))^{1}$
$\otimes$	$N_2^{SA3:SA}$ defeats N_1^{Inst}
$\mathbb{NS}'$	$Ifact^{SA3:SA}(Order(Ref:4, From:Sam, To:Tom, Product:P3, Quantity:5))^{1}$
	$Obl_{Tom}^{SA3:SA}(Delivery(Ref:4, From:Tom, To:Sam, Product:P3, Quantity:5) \prec 3)$
$\mathbb{NS}$	$Ifact^{SA3:SA}(Order(Ref:4, From:Sam, To:Tom, Product:P3, Quantity:5))^{1}$
	$Obl_{Tom}^{SA3:SA}(Delivery(Ref:4, From:Tom, To:Sam, Product:P3, Quantity:5) \prec 3)$
	$Ifact^{SA3:SA}(Delivery(Ref:4, From:Tom, To:Sam, Product:P3, Quantity:5))^{2}$
	$Fulf^{SA3:SA}(Obl_{Tom}^{SA3:SA}(Delivery(Ref:4, From:Tom, To:Sam, Product:P3, Quantity:5) \prec 3))^{2}$
$\otimes$	none, $N_3^{SA3:SA}$ applies
$\mathbb{NS}'$	$Ifact^{SA3:SA}(Order(Ref:4, From:Sam, To:Tom, Product:P3, Quantity:5))^{1}$
	$Obl_{Tom}^{SA3:SA}(Delivery(Ref:4, From:Tom, To:Sam, Product:P3, Quantity:5) \prec 3)$
	$Ifact^{SA3:SA}(Delivery(Ref:4, From:Tom, To:Sam, Product:P3, Quantity:5))^{2}$
	$Fulf^{SA3:SA}(Obl_{Tom}^{SA3:SA}(Delivery(Ref:4, From:Tom, To:Sam, Product:P3, Quantity:5) \prec 3))^{2}$
	$Obl_{Sam}^{SA3:SA}(Payment(Ref:4, From:Sam, To:Tom, Amount:5) \prec 4)$

5.7 Discussion

The research work described in this chapter has evolved through a number of publications. A preliminary hierarchical model of a normative framework supporting norm inheritance can be found in [136], where a three layered approach was followed. In this case, aiming at Virtual Enterprise settings, we have identified an *institutional* layer for predefined "default rules", a *constitutional* layer for specifying norms that will regulate a particular VE, and an *operational* layer that includes instantiations of constitutional norms as the operation of the VE takes place. A first step towards using contexts for framing norms is in [135]. A model for institutional reality, together with a formalization of rules and norms for providing a dynamic normative environment was presented in [139], and a more practical account was given in [137]. Finally, the generalization of the three-layered approach outlined above into the model as presented in this chapter can be found in [140] and [141].

5.7.1 Environment specification

In general, the notion of multi-agent systems assumes the existence of a common *environment*, where agent interactions take place. Recently more attention has being given to the environment as a first-class entity [234][233]. Electronic institutions provide an environment whose main task is to support governed interaction by maintaining the normative state of the system, embracing norms that apply to each of the interacting agents. An EI can thus be seen as an interaction-mediation infrastructure (as discussed by Weyns *et al.* [233]. Other researchers have made a deeper analysis of electronic institutions as environment engineering tools, namely in [10] and [220].

Our approach to create institutional reality resembles the notion of "influence and reaction" as proposed by Ferber and Müller [74]. In our case, influences comprise agents' illocutions (brute facts), through which they try to modify the state of the normative environment, trying to convince the EI that certain events took place. The environment then reacts to such influences by applying the constitutive and institutional rules (the "laws of the world" [74]) and producing elements of institutional reality. However, ours is an asynchronous action model, since agents can run asynchronously and independently of the environment itself (closer to the model by Weyns and Holvoet [232]). The normative environment comprises an active service that can change the state of the system independently of agents' actions [170]. For instance, a violation can be caused by the absence of an institutional fact (which denotes the absence of an appropriate agent action) at a certain deadline. Of course this approach assumes the existence of a synchronized clock mechanism for every agent.

One of the most important principles of our approach is the assumption of a non-static normative environment; this means that we depart from a more conservative view of norms seen as a set of preexistent interaction conventions that agents are willing to comply with (as in the *adscription* approach in [10]). We pursue an EI that provides a supportive normative framework whose main purpose is to facilitate the establishment of further commitments among a group of contracting agents.

The possibility of having an underlying normative framework, from which norms may be inherited, is a distinguishing feature of our approach, as is the "loose coupling" between norms and contrary-to-duties. Also, the institution includes norm monitoring policies that span all created contracts. This is in contrast with other approaches, namely [227], where these policies and repair measures are spread among the norms themselves. This prevents the use of inheritance mechanisms that make the normative framework more flexible. Our norm formalism allows us to implement norm monitoring as a context-independent activity. Specifically, we distinguish violation detection from sanction imposition. While the detection of violations is a general and institutionally defined concept, the prescription of sanctions may be contract-specific.

López-y-López and Luck [149] presented a theoretical norm schema that includes, besides normative goals, preconditions and addressees, also rewards and punishments. While in order to be effective norms do need such elements (e.g. an obligation needs punishments), within a normative framework rewards and punishments are enforcement measures that may be institutionally defined. We adopt a simpler and more modular norm specification, keeping sanctions apart of each norm definition. The specification of norms in a contract often configures a chaining of obligations, which resembles the notion of *interlocking norms* in [149]: norms that get activated through the fulfillment or violation of other norms.

García-Camino *et al.* [87] present a grammar for rules that combines both our rule and norm definitions. However, our concern is to distinguish *a priori* rule definition as a normative state maintenance issue from norm definition as a contracting activity.

A recent proposal to define a programming language for programing normative artifacts has been made by Tinnemeier *et al.* [215], which incorporates several notions that we have included in our model. Norms define states of the world to be achieved, and are as such "declarative" (as opposed to "procedural" norms referring to actions). Specific elements are used to express brute and institutional facts, as well as violations.

We regarded (institutional) roles, namely those defined for trusted-third parties, strictly as a means to define which agents are empowered to create institutional facts. However, the notion of role has typically a set of norms attached that agents enacting the role are expected to follow. V. Dignum [64] considers that a role-enacting agent has a *social contract* attached, which may include a set of contract clauses (defined as deontic expressions). In the case of OperA, a role description defined at the organizational model may be inherited into a social contract, which therefore need not specify any additional clauses. The same approach could be adopted for our institutional roles: an agent enacting a role could be seen as establishing a social contract with the EI, thereby creating a context that would inherit predefined norms describing how the enacting agent should behave.

5.7.2 Institutional facts and normative positions

Several researchers have taken inspiration from Searle's theory of constitutive rules [202][203], as well as from Jones and Sergot's institutionalized power [117]. In our case, we worked on the concept of institutional reality in order to bring into an institutional environment a mapping of real-world events recognized by the EI. Constitutive rules, by defining "counts-as" relations, specify empowerments [117] (also called *authorizations* in [79]) of agents performing specific roles, which are thus seen by the EI as trusted third parties. Given that our main intention in defining such roles is not to regulate how agents enacting them should behave, but instead to grant them a privileged position concerning the creation of institutional facts, we assume that agents are permitted to make their authorized statements.

Knottenbelt [126][127] uses a model of authorization that enables agents participating in a contract to initiate event calculus fluents pertaining to contract enactment. The definition of *authoritative agents* specifies how informational messages affect the contract state. The same mechanism is used to identify trusted-third parties, by making them authoritative regarding certain events (e.g. bank for payments, delivery tracker for deliveries). A non-authorized attempt will have no effect in the contract state (that is, will not initiate any fluent). This approach is similar to ours: informative illocutions which do not have associated constitutive rules will have no effect in producing institutional facts. However, we also included the possibility of defining constitutive rules that require a combination of statements made by different agents, and illustrated this approach by relying directly on the parties involved in a specific exchange.

García-Camino *et al.* [87] include in their concept of institutional rules the validation of attempts to utter illocutions. Such attempts become "legal utterances" if they are permitted. Conceptually, it seems that in this case agents cannot even make statements that they are not allowed to do, although they might be punished for attempting to make them. This is due to the fact that their approach in modeling electronic institutions, through a performative structure based on scenes, is highly dialogical. An emphasis is thus put in what agents are allowed to utter in each scene.

Modgil *et al.* [162] propose a monitoring framework that includes *observers* as entrusted entities that accurately report the behaviors that they observe from contractual agents. This is quite similar to Jones and Sergot's model of institutionalized power [117] that we incorporated in our own approach.

Norms are typically related with the deontic notions of obligation, permission and prohibition. However, in the contracting domain obligations are of primary importance. Legal theorists define contract law as being part of the law of obligations, "concerned with obligations that people owe to others as a result of the relations and transactions in which they become involved" ([16], p. 1). Contract law puts an emphasis on obligations incurred through promises that parties make with respect to some action to be performed or state of affairs to bring about. For this reason, we essentially rely on obligations to specify contractual commitments. Contractual rights are dealt with in an implicit way: norms can be triggered with institutional facts, which therefore can be used to exercise a right of demanding a certain contribution from a contractual partner.

Our approach to monitor obligations assumes that it is in the best interest of contractual partners to inform the institution of their abidance to obligations. Detecting violations of prohibitions is much harder, because it demands for a pervasive character of the institution; actions may be performed which are not observable by the enforcing entity (Vázquez-Salceda *et al.* [227] study different levels of verifiability).

Although we are primarily concerned with deadline obligations, the inclusion of permissions or prohibitions as deontic statements prescribed by norms would demand no changes in our norm activation defeasibility approach. We do not rely on conflicts between the content of deontic statements (which are deontic conflicts), but instead on norm activation conflicts.

5.7.3 Context

The notion of *context* as used in our approach seeks to capture the notion of "business context" that seems to be present in many B2B contractual situations. Furthermore, Searle's constitutive rules have contexts attached that we believe could be seen as contractual contexts as we model them. This would enable us to iterate through institutional facts inside specific contractual relations: each contract gives birth to a context within which certain constitutive rules would apply. While we did not include this extension in the model as presented in this chapter (but have considered it before [139][138]), we believe that it makes sense because it allows contract fulfillment to be adjusted by matters of trust between contractual partners or due to business specificities – in a specific contract an institutional fact can count as another institutional fact. This approach also opens up the possibility of extending the ontology for institutional facts.

The idea of context for normative reasoning has been studied before. However, in most cases the notion of context comes from the 'counts-as' relation [203][117]: "X counts-as Y in context C". For instance, in [228][103] a context gives an interpretation to abstract norms of a broader context. There is a leveled structuring of contexts, which broadly contemplates institutions, sub-institutions and organizations, from the most abstract to the most concrete level. However, concrete norms (refined as rules and implemented as procedures) are used to model preexistent organizations. Concept abstraction is studied in [104]. In this case, it is not the norm that is abstract, but instead the concepts in which it is expressed. A norm based on abstract concepts may be further specified in a more specific context. Our approach has a different concern: we use the context structure for designing a model of defeasibility for norms, which may be added to the system at runtime. We do not tackle with abstraction.

The "contextualization" of contracts within higher normative structures has also been advanced in [38]. In this case, a contract is modeled as an institution itself (see also [21]), and can be governed by another (super) institution. This relationship is expressed through a mechanism of empowerment. States are described by fluents and evolve according to rules expecting events. Empowerments are defined by normative fluents allowing the creation of events and the initialization or termination of fluents. With this approach, a rule defined in an institution may operate on another institution's state if the rule's effects are explicitly empowered. In our approach, contracts are modeled as contexts within a single institution. Norms can also operate in contexts other than the one where they are defined, but this property is based on a structured normative framework, and not on a discretionary basis that may be cumbersome to express.

5.7.4 Normative conflict handling

From the law field, three normative conflict resolution principles have been defined and are traditionally being used. The *lex superior* is a hierarchical criterion implying that a norm issued by a more important legal entity prevails, when in conflict with another norm (e.g. the Constitution prevails over any other legal body). The *lex posterior* is a chronological criterion indicating that the most recent norm prevails. The *lex specialis* is a specificity criterion establishing that the most specific norm prevails. While not firmly adopting any of these options, our approach resembles more the *lex specialis* principle. However, the defeating norms are more specific in the sense that they are *defined at* (as opposed to *applied to*) a more specific context (a kind of "lex inferior"). The *lex specialis* flavor comes from the fact that in most cases a defeating norm should apply to a narrower context-set. These properties of our norm defeasance approach result from the fact that the original aim is not to impose predefined regulations on agents, but instead to help them in building contractual relationships by providing a normative background (which can be exploited in a partial way through adaptation). A feature of our approach that exposes this aim is that all norms are defeasible. In this respect we follow the notion from law theory of "default rules" [48], which should be seen as facilitating rather than constraining contractual activity [118].

This notion of "default rules" might be misleading; it has not a direct correspondence with default logic formalizations [184]. We do not handle the defeasibility of conclusions of default rules in that sense, but instead model defeasibility of the application of the rules themselves (which are called norms).

From a theoretical logical stance, norm defeasibility has been addressed in, e.g., [192][198][221]. Typically, deontic reasoning [169] guides these approaches, and thus conflicts regard the deontic operators themselves. Our approach is centered instead on the applicability of norms, not on their conclusions.

The work by García-Camino *et al.* in [86] addresses the issue of conflict resolution in a structured setup of compound activities. These resemble our context and sub-context relationships. However, those authors model deontic conflicts (e.g. an action being obliged and prohibited), while we model norm (activation) conflicts. They study the inheritance of normative positions (obligations, permissions, prohibitions), based on an explicit stamping of each one of them with a priority value and a timestamp; the specificity criterion is based on the compound activities' structure. We address the inheritance of norms and provide a means to override norm activations based on their defeasibility.

Our approach of context and sub-context definitions, together with the norm defeasibility model presented, is similar to the notion of *supererogatory defeasibility* in [194]. They model defeasibility in terms of role and sub-role definitions. In fact, they also consider *express defeasibility*, which is based on the specificity of conditions for norm applicability, but this approach has been followed by several others.

The problem of normative conflict resolution has been also addressed through more practical approaches (e.g. in the B2B domain). The application of business rules in e-commerce has been addressed in [98], where courteous logic programs allow for an explicit definition of priorities among rules. An extension based on defeasible [168] and deontic logic has been advanced in [94] for the representation of business contracts (and not merely business rules). However, this approach does not consider defeasibility of norms between a contract and an underlying normative framework. Finally, [89] also addresses defeasible reasoning in the e-contracts domain, based on the translation of contracts from event calculus to default logic, and on the definition of dynamic priorities among rules (by using domain-dependent criteria). Conflicts are, in this case, based on the normative positions of agents.

Finally, we assume a close connection between norm activation conflicts as we model them and the notion of *conflict set* (or *agenda*) in rule-based forward-chaining systems (e.g. [82]). In those systems, a conflict arises whenever there is a possible application of more than one rule at the same time, and a conflict resolution strategy will decide which rule to apply in each step of the reasoning process. In our case, conflict resolution is based on hierarchical relationships between the contexts where norms are defined.

Chapter 6

Revisiting Contractual Obligations

The essence of contract is commitment [156]: contracts provide a legally binding agreement including legal sanctions in case of failure to honor commitments. Nevertheless, real-world business relationships have an essentially cooperative nature. Therefore, the importance of successfully proceeding with business demands for flexibility of operations: contractors should try to facilitate the compliance of their partners. This common goal of conducting business is based on the fact that group success also benefits each partner's private goals. These goals are not limited to the ongoing business relationship, but may also concern future opportunities that may arise.

This observation renders inappropriate the usual semantics of deadline obligations as used in MAS (for which a representative source is [62], which we also followed in Chapter 5): deadline obligations are violated if the obliged action or state is not obtained until the deadline is reached. The problem is that this generic statement is too rigid, and abstracted away from a potential application domain. We argue that in some domains – such as in business contracts – this approach is not desirable.

In this chapter an alternative view to address contractual obligations is proposed, which will also imply changes in the elements that compose the normative state. We will provide a formalization, based on temporal logic, of our approach to model contractual obligations. Afterwards we will adequate the use of institutional rules to this new semantics, which will then enable us to provide an unambiguous implementation in Chapter 7.

6.1 Obligations in Contracts

Contract law treats obligations in contracts as distinct from other kinds of obligations. Atiyah [16] points the following particularities of contractual obligations (also analyzed in [50]):

- they have a private character, in the sense that they apply only to those parties that have assumed them;
- they are owed by individuals to other individuals specifically, rather than the public in general;
- they are enforceable only by the persons to whom they are owed, i.e., no public authority will take initiative to enforce the contract;
- they are self-imposed, i.e., they are obligations arising from an agreement, promise, or other undertaking.

From this characterization we emphasize two important points. First, contractual obligations become applicable only when individuals commit to them through a process of contract establishment. Second, contractual obligations are directed from a contractual party to another contractual party.

This last observation has lead to the notion of *directed obligations* [110]: obligations seen as directed from a *bearer* (responsible for fulfilling the obligation) to a *counterparty*. In fact, some researchers, such as Ryu [194], define *contractual obligation* as an obligation with an "obligor" (bearer) and an "obligee" (counterparty). The relationship between these two roles in a directed obligation has been studied, giving rise to two different theories. The *benefit theory* promotes the fact that the counterparty of an obligation is intended to benefit from its fulfillment (see [110] for a benefit theory perspective of directed obligations). A more relevant approach, at least in which contract enforcement is concerned – the *claimant theory* – takes the stance that obligations are interpreted as claims from counterparties to bearers (see [212] for a claimant theory support).

In general, claimant approaches are based on the following definition for directed obligation (adapted from [212]):

$$Obl_{b,c}(f) =_{def} Obl_b(f) \wedge (\neg f \Rightarrow Perm_c(la_b))$$

A directed obligation from agent b towards agent c to bring about f means that b is obliged to bring about f and if b does not bring about f then c is *permitted* to initiate legal action against b. The concept of legal action is, however, rather vague. A similar approach is taken in [59], where agent c is said to be *authorized* to repair the situation in case b does not fulfill his obligation. Repair actions include demanding further actions from b; e.g., c may demand compensation for damages. It is interesting to note that such definitions are careful enough to base the claims of the counterparty on the *non-fulfillment* of the obligation, not on its violation. In fact, these definitions do not include deadlines, which are often the basis for violation detection. Another significant issue is the discretionary nature of the counterparty's reaction (he is permitted or authorized), instead of an automatic response based on the non-fulfillment of the bearer (as would be the case in automatic violation detection approaches based on deadlines, complemented with the definition of violation reaction norms).

6.2 Directed Obligations with Flexible Time Windows

In this section we propose a new model for contractual obligations that is based on combining deadline obligations with directed obligations, obtaining *directed deadline obligations*. While presenting our model, we will make use of a real-world legislation on trade contracts, namely the United Nations Convention on Contracts for the International Sale of Goods (CISG) [217]. Among several other issues, this convention establishes what parties may do in case of deadline violations. In some cases they are allowed to fulfill their obligations after the deadline (Article 48), or to extend the deadlines with the allowance of their counterparties. Furthermore, a party may extend his counterparty's deadlines (Articles 47 and 63), which denotes a flexible and even collaborative facet of trade contracts.

While using the CISG convention as a source for modeling contractual obligations, we believe that the features we are trying to extract from this legislation are general enough to be representative of any kind of contractual relationship.

6.2.1 Directed deadline obligations

Our proposal combines directed [110][212] and deadline [27] obligations, in order to obtain a more precise definition of when it is that a counterparty may claim against the inability of a bearer to fulfill the obligation. We will motivate and formalize the notion of *directed deadline obligation* – $Obl_{b,c}(f \prec d)$: agent b is obliged towards agent c to bring about f before d. An extension of directed (contractual) obligations with temporal restrictions is also introduced in [194], but that approach is based on a rigid model of violations, in that they are automatically obtained at the deadline. In our approach deadlines have a distinct role in the semantics of obligations. We will introduce the notion of *deadline violation* (as opposed to obligation violation) in order to obtain a flexible approach to handle non-ideal situations: each deadline violation is different, since each may have a different impact on the ongoing business; furthermore, each deadline violation occurs between a specific pair of agents with a unique trust relationship.

In fact, deadline handling is central to define the semantics of contractual obligations. Looking at the CISG convention [217], we have:

> Article 48: (1) [...] the seller may, even after the date for delivery, remedy at his own expense any failure to perform his obligations, if he can do so without unreasonable delay [...]; (2) If the seller requests the buyer to make known whether he will accept performance and the buyer does not comply with the request within a reasonable time, the seller may perform within the time indicated in his request. [...]

This means that even though a deadline has been violated, the bearer may still be entitled to fulfill *the same* obligation. This kind of delay is also called a

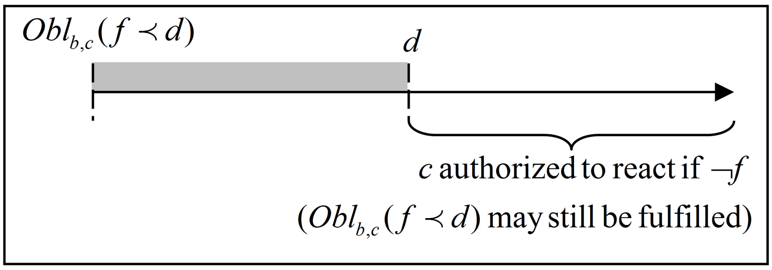


Figure 6.1: Directed obligation with deadline.

grace period: a period beyond a due date during which an obligation may be met without penalty or cancellation.

Figure 6.1 illustrates the intuitive semantics of directed deadline obligations. The shaded area represents the period of time within which the achievement of f will certainly bring a fulfillment of the obligation. The region to the right of d indicates that counterparty c is entitled to react if f is not accomplished; however, as long as no reaction is taken, b can still fulfill his obligation.

Therefore, a deadline violation brings a counterparty authorization. Authorizations are taken into account in the normative system by having rules and norms that are based on the materialization of such authorizations.

6.2.2 Livelines and deadlines

The deadline approach is often taken to be appropriate for specifying temporal restrictions on obligations. However, in certain cases a time window should be provided. In international trade transactions, for instance, storage costs may be relevant. Also, perishable goods should be delivered only when they are needed, not before. This is why in CISG [217] we have:

> *A*rticle 52: (1) If the seller delivers the goods before the date fixed, the buyer may take delivery or refuse to take delivery.

Therefore, anticipated fulfillments are not always welcome. We find it necessary to include a variation of directed deadline obligations, to which we add a *liveline*: a time reference after which the obligation should be fulfilled. In this case we have $Obl_{b,c}(l \prec f \prec d)$: agent b is obliged towards agent c to bring about f between l (a liveline) and d (a deadline). Figure 6.2 illustrates the intuitive semantics of this kind of obligation. The shaded area represents the period of time within which the achievement of f will certainly bring a fulfillment of the obligation. If f is accomplished before l, however, it may be the case that c is not willing to accept such a fulfillment, or at least that he may not be happy about it – the region to the left of l entitles c to react if f is accomplished. The region to the right of d is as with (simple) directed deadline obligations.

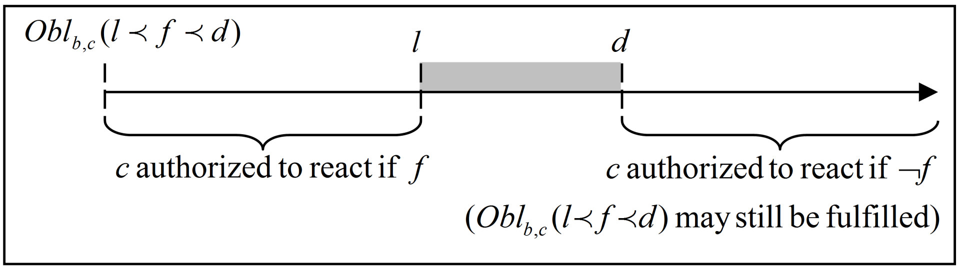


Figure 6.2: Directed obligation with liveline and deadline.

We escape from an approach with a fixed time reference for obligation fulfillment (an obligation for bringing about f at time t), which would be suggested by the term "date fixed" in CISG's Article 52 transcription above. We find it more convenient to define a fixed date as an interval, say, from the beginning till the end of a specific date.

6.2.3 Handling liveline and deadline violations

After we have advocated a counterparty authorization approach to temporal violations, in this section we analyze the kind of actions that the counterparty may take in such situations.

The successful enactment of a contract is dependent on the need to make contractual provisions performable in a flexible way. The importance of having flexible trade procedures is apparent, once again, in the CISG convention [217]:

> *A*rticle 47: (1) The buyer may fix an additional period of time of reasonable length for performance by the seller of his obligations.

> *A*rticle 63: (1) The seller may fix an additional period of time of reasonable length for performance by the buyer of his obligations.

These articles emphasize, once more, the need for flexible deadlines. Note that the counterparty's benevolence on conceding an extended deadline to the bearer does not prescribe a new obligation; instead, *the same* obligation may be fulfilled within a larger time window. Furthermore, it is also in the counterparty's best interest that this option is available, given the importance of reaching success in the performance of the contract.

In some other cases, a party may decide that the non-fulfillment of an obligation should be handled in a more strict way. The CISG convention specifies conditions for cancelling a contract in case of breach:

> *A*rticle 49: (1) The buyer may declare the contract avoided: (a) if the failure by the seller to perform any of his obligations [...] amounts to a fundamental breach of contract; [...]; (2) However, in cases where the seller

> has delivered the goods, the buyer loses the right to declare the contract avoided unless he does so: (a) in respect of late delivery, within a reasonable time after he has become aware that delivery has been made; [...]
>
> *A*rticle 64: (1) The seller may declare the contract avoided: (a) if the failure by the buyer to perform any of his obligations [...] amounts to a fundamental breach of contract; [...]; (2) However, in cases where the buyer has paid the price, the seller loses the right to declare the contract avoided unless he does so: (a) in respect of late performance by the buyer, before the seller has become aware that performance has been rendered; [...]

These articles allow contract termination in both non-performance and late performance cases. However, the second case is limited to the awareness of the offended party.

From these excerpts we can distinguish two kinds of reactions to non fulfillments: a smoother one (from Articles 47, 48 and 63), in which parties are willing to recover from an initial failure to conform to an obligation; and a stricter one (Articles 49 and 64), where the failure is not self-containable anymore. Based on these options, we propose a model for a directed deadline obligation lifecycle.

6.2.4 Modeling the semantics of directed obligations with time windows

Following the discussion above, in Table 6.1 we identify the possible states for an obligation, together with the elements[1] we shall use to signal some of those states (when obtained, these elements are supposed to persist over time).

We now proceed to formalize each type of obligation. As with deadline obligations in the previous chapter, we will make use of linear temporal logic.

Directed deadline obligations

Figure 6.3 illustrates, by means of a state transition diagram, the lifecycle of directed deadline obligations. We take obligations as being prescribed from conditional norms; the confirmation of the norm's conditions will change the prescribed obligation's state from inactive to active. The obligation is also automatically pending, since it may be legitimately fulfilled right away. We set the obligation to have a violated deadline – $DViol(obl)$ – when the deadline occurs before the obliged fact.

The counterparty's reaction to a deadline violation will only change the obligation's state if the option is to deem the obligation as violated, by *denouncing* this situation. For this we introduce the element $Den_{c,b}(obl)$, which is a denounce from agent c towards agent b regarding the failure of the latter to comply with his

[1]For simplification, context references are omitted.

Table 6.1: States of directed obligations with time windows.

inactive	the obligation is not yet in effect, but will eventually be prescribed by a norm
active: $Obl_{b,c}(f \prec d)$ or $Obl_{b,c}(l \prec f \prec d)$	the obligation was prescribed by a norm
pending	the obligation may be fulfilled from now on
liveline violation: $LViol(obl)$	the fact being obliged in *obl* has been brought ahead of time
deadline violation: $DViol(obl)$	the fact being obliged in *obl* should have been brought already
fulfilled: $Fulf(obl)$	obligation *obl* was fulfilled
violated: $Viol(obl)$	obligation *obl* was violated and cannot be fulfilled anymore

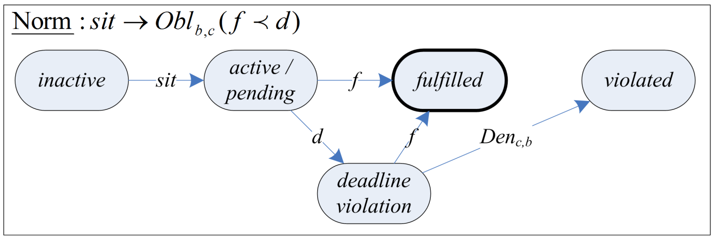


Figure 6.3: Lifecycle of a directed deadline obligation.

obligation *obl*. Denounces may be obtained through a constitutive rule based on message deliveries, as shown in Section 5.4.2. Since we consider the achievement of facts to be common knowledge, a party may only denounce the non-fulfillment of an obligation while the obliged fact is not obtained[2].

Formally, we start by identifying the (absolute) fulfillment case:

$$Obl_{b,c}(f \prec d) \wedge (Ifact(f)\ B\ Time(d)) \models Fulf(Obl_{b,c}(f \prec d)) \qquad (6.1)$$

Then we state the consequence of reaching a deadline with no achievement of the obligated fact:

$$Obl_{b,c}(f \prec d) \wedge (Time(d)\ B\ Ifact(f)) \models DViol(Obl_{b,c}(f \prec d)) \qquad (6.2)$$

Note that, differently from the usual treatment of deadline obligations (such as [27] and the preliminary approach that we have shown in the previous chapter), we set the obligation to have a violated deadline – $DViol(Obl_{b,c}(f \prec d))$ – but not to be violated in itself (i.e., the obligation may still be fulfilled).

[2] This is a simplification of what Articles 49 and 64 of CISG suggest.

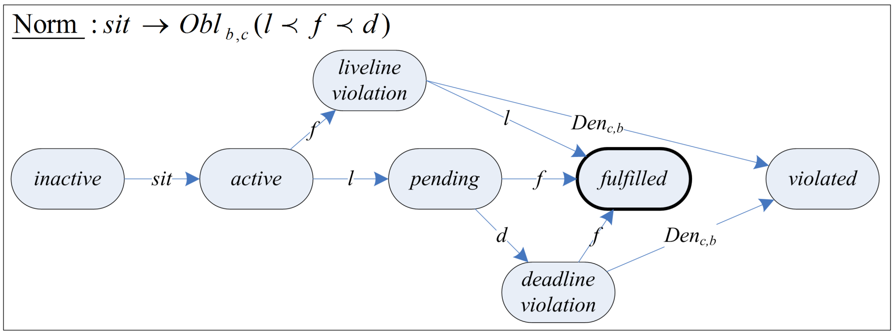


Figure 6.4: Lifecycle of a directed obligation with liveline and deadline.

Finally, we identify two possible outcomes from the violated deadline case:

$$DViol(Obl_{b,c}(f \prec d)) \wedge (Ifact(f)\ B\ Den_{c,b}(Obl_{b,c}(f \prec d))) \models Fulf(Obl_{b,c}(f \prec d)) \tag{6.3}$$

$$DViol(Obl_{b,c}(f \prec d)) \wedge (Den_{c,b}(Obl_{b,c}(f \prec d))\ B\ Ifact(f)) \models Viol(Obl_{b,c}(f \prec d)) \tag{6.4}$$

Directed obligations with liveline and deadline

Figure 6.4 shows the state transition diagram for directed obligations with liveline and deadline. In this case, the obligation will only be pending when l arises, since only then it may be fulfilled in a way that is compliant with the terms of the contract. We have now two kinds of temporal violations: liveline violations of the form $LViol(obl)$ and deadline violations of the form $DViol(obl)$. In both cases, a denounce ($Den_{c,b}(obl)$) may establish the obligation as violated, if issued before l or f, respectively.

The lifecycle of directed obligations with liveline and deadline is formalized as follows:

$$Obl_{b,c}(l \prec f \prec d) \wedge (Ifact(f)\ B\ Time(l)) \models LViol(Obl_{b,c}(l \prec f \prec d)) \tag{6.5}$$

$$LViol(Obl_{b,c}(l \prec f \prec d)) \wedge (Time(l)\ B\ Den_{c,b}(Obl_{b,c}(l \prec f \prec d))) \models Fulf(Obl_{b,c}(l \prec f \prec d)) \tag{6.6}$$

$$LViol(Obl_{b,c}(l \prec f \prec d)) \wedge (Den_{c,b}(Obl_{b,c}(l \prec f \prec d))\ B\ Time(l)) \models Viol(Obl_{b,c}(l \prec f \prec d)) \tag{6.7}$$

$$Obl_{b,c}(l \prec f \prec d) \wedge (Time(l)\ B\ Ifact(f)) \wedge (Ifact(f)\ B\ Time(d)) \models Fulf(Obl_{b,c}(l \prec f \prec d)) \tag{6.8}$$

$$Obl_{b,c}(l \prec f \prec d) \wedge (Time(d)\ B\ Ifact(f)) \models DViol(Obl_{b,c}(l \prec f \prec d)) \tag{6.9}$$

$$\begin{aligned} DViol(Obl_{b,c}(l \prec f \prec d)) \wedge (Ifact(f)\ B\ Den_{c,b}(Obl_{b,c}(l \prec f \prec d))) \\ \models Fulf(Obl_{b,c}(l \prec f \prec d)) \end{aligned} \tag{6.10}$$

$$\begin{aligned} DViol(Obl_{b,c}(l \prec f \prec d)) \wedge (Den_{c,b}(Obl_{b,c}(l \prec f \prec d))\ B\ Ifact(f)) \\ \models Viol(Obl_{b,c}(l \prec f \prec d)) \end{aligned} \tag{6.11}$$

6.2.5 Smoother authorizations on violations

The diagrams in figures 6.3 and 6.4 only include events that produce a change in an obligation's state. The denouncement of the non-fulfillment of an obligation, making it violated and consequently not fulfillable any longer, denotes a situation in which a bearer's attempt to fulfill the obligation will no longer be significant to the counterparty, and thus a consummated violation should be handled according to applicable norms. These may bring sanctions, further obligations or ultimately a contract cancellation, as in Articles 49 and 64 of CISG.

In order to accommodate less strict situations, we consider that in liveline and deadline violation states, while the obligation can still be fulfilled, the counterparty may react to the non-ideal situation. These possibilities are not illustrated in figures 6.3 and 6.4 because they do not bring state changes. For instance, in international trade transactions storage costs may be relevant. The counterparty may therefore be authorized to demand for payment of storage costs from an early compliant bearer. Another example for the deadline violation case:

> *A*rticle 78: If a party fails to pay the price or any other sum that is in arrears, the other party is entitled to interest on it [...]

While obligation state transitions are processed with appropriate rules (including rules that take denounces into account), authorizations expressing the counterparty's right to demand for compensation are handled by the system through appropriate norms, which may be defined in a contract basis.

6.3 Monitoring Contractual Obligations with Institutional Rules

Now that we have a new model for contractual obligations, in this section we reimplement their semantics through institutional rules. We will start by providing an essentially direct mapping from the semantics provided in Section 6.2.4.

In order to detect the moment at which the *before* relation holds, we translate terms of the form $(e_1\ B\ e_2)$ into a conjunction $e_1 \wedge \neg e_2$. For directed obligations with liveline and deadline, we obtain the following rules[3] (which correspond to formulae (6.5) to (6.11)):

$$Obl_{b,c}(l \prec f \prec d) \wedge Ifact(f) \wedge \neg Time(l) \rightarrow LViol(Obl_{b,c}(l \prec f \prec d)) \tag{6.12}$$

$$\begin{aligned} LViol(Obl_{b,c}(l \prec f \prec d)) \wedge Time(l) \wedge \neg Den_{c,b}(Obl_{b,c}(l \prec f \prec d)) \\ \rightarrow Fulf(Obl_{b,c}(l \prec f \prec d)) \end{aligned} \tag{6.13}$$

$$\begin{aligned} LViol(Obl_{b,c}(l \prec f \prec d)) \wedge Den_{c,b}(Obl_{b,c}(l \prec f \prec d)) \wedge \neg Time(l) \\ \rightarrow Viol(Obl_{b,c}(l \prec f \prec d)) \end{aligned} \tag{6.14}$$

$$\begin{aligned} Obl_{b,c}(l \prec f \prec d) \wedge Time(l) \wedge \neg LViol(Obl_{b,c}(l \prec f \prec d)) \wedge Ifact(f) \wedge \\ \neg Time(d) \rightarrow Fulf(Obl_{b,c}(l \prec f \prec d)) \end{aligned} \tag{6.15}$$

$$Obl_{b,c}(l \prec f \prec d) \wedge Time(d) \wedge \neg Ifact(f) \rightarrow DViol(Obl_{b,c}(l \prec f \prec d)) \tag{6.16}$$

$$\begin{aligned} DViol(Obl_{b,c}(l \prec f \prec d)) \wedge Ifact(f) \wedge \neg Den_{c,b}(Obl_{b,c}(l \prec f \prec d)) \\ \rightarrow Fulf(Obl_{b,c}(l \prec f \prec d)) \end{aligned} \tag{6.17}$$

$$\begin{aligned} DViol(Obl_{b,c}(l \prec f \prec d)) \wedge Den_{c,b}(Obl_{b,c}(l \prec f \prec d)) \wedge \neg Ifact(f) \\ \rightarrow Viol(Obl_{b,c}(l \prec f \prec d)) \end{aligned} \tag{6.18}$$

Rule 6.15 demanded a more careful construction, since we have, in formula 6.8, two consecutive *before* relations – we need to ensure that there is no liveline violation when both $Time(l)$ and $Ifact(f)$ are true.

6.3.1 Reasoning with time

In Section 5.5.2 we have already acknowledged the benefits of time-stamping events, namely the possibility of relaxing the rule evaluation policy: rules need no longer be evaluated at every normative state update. Furthermore, in business contracts it is common to have deadlines that are dependent on the fulfillment date of other obligations. Therefore, instead of having fixed (absolute) dates, these may at times be relative, calculated according to other events. CISG [217] expresses this by saying that dates can be *determinable* from the contract:

> *A*rticle 33: The seller must deliver the goods: (a) if a date is fixed by or determinable from the contract, on that date; (b) if a period of time is fixed by or determinable from the contract, at any time within that period [...]

[3]The simpler case of directed deadline obligations is a simplification over these rules.

Table 6.2: Institutional reality elements.

$\mathit{Ifact}(f)^t$	:	fact f is institutionally recognized as having occurred at time t
$\mathit{Time}(t)$	:	instant t has elapsed
$\mathit{Obl}_{b,c}(f \prec d)^t$	:	agent b is obliged, since t, towards agent c to bring about f until d
$\mathit{Obl}_{b,c}(l \prec f \prec d)^t$	:	agent b is obliged, since t, towards agent c to bring about f between l and d
$\mathit{LViol}(\mathit{obl})^t$	:	there was a liveline violation of obligation obl at time t
$\mathit{DViol}(\mathit{obl})^t$	:	there was a deadline violation of obligation obl at time t
$\mathit{Fulf}(\mathit{obl})^t$	:	obligation obl was fulfilled
$\mathit{Viol}(\mathit{obl})^t$	:	obligation obl was violated
$\mathit{Den}_{c,b}(\mathit{obl})^t$	:	agent c has denounced, at time t, the failure of agent b to fulfill obligation obl

> *A*rticle 59: The buyer must pay the price on the date fixed by or determinable from the contract [...]

It is therefore useful to timestamp each event, that is, institutional reality elements. Rules and norms will make use of these temporal references. In Table 6.2 we present a modified version of institutional reality elements. As compared with Def. 5.4.1, for simplification we did not include a context reference in these elements (within rules the context of each element is assumed to be the same).

6.3.2 Re-implementing rules

Now that events include a timestamp, we are able to re-implement monitoring rules by taking advantage of this feature. We also need our rules to assign timestamps to the institutional reality elements that they add to the normative state. The following institutional rules implement the semantics of directed obligations with liveline and deadline:

$$\mathit{Obl}_{b,c}(l \prec f \prec d)^i \wedge \mathit{Ifact}(f)^t \wedge t < l \rightarrow \mathit{LViol}(\mathit{Obl}_{b,c}(l \prec f \prec d)^i)^t \tag{6.19}$$

$$\begin{aligned} \mathit{LViol}(\mathit{Obl}_{b,c}(l \prec f \prec d)^i)^j \wedge \mathit{Time}(l) \wedge \neg(\mathit{Den}_{c,b}(\mathit{Obl}_{b,c}(l \prec f \prec d)^i)^u \wedge u < l) \\ \rightarrow \mathit{Fulf}(\mathit{Obl}_{b,c}(l \prec f \prec d)^i)^l \end{aligned} \tag{6.20}$$

$$\begin{aligned} \mathit{LViol}(\mathit{Obl}_{b,c}(l \prec f \prec d)^i)^j \wedge \mathit{Den}_{c,b}(\mathit{Obl}_{b,c}(l \prec f \prec d)^i)^u \wedge u < l \\ \rightarrow \mathit{Viol}(\mathit{Obl}_{b,c}(l \prec f \prec d)^i)^u \end{aligned} \tag{6.21}$$

$$\mathit{Obl}_{b,c}(l \prec f \prec d)^i \wedge \mathit{Ifact}(f)^t \wedge l < t \wedge t < d \rightarrow \mathit{Fulf}(\mathit{Obl}_{b,c}(l \prec f \prec d)^i)^t \tag{6.22}$$

$$Obl_{b,c}(l \prec f \prec d)^i \wedge Time(d) \wedge \neg(Ifact(f)^t \wedge t < d) \\ \rightarrow DViol(Obl_{b,c}(l \prec f \prec d)^i)^d \tag{6.23}$$

$$DViol(Obl_{b,c}(l \prec f \prec d)^i)^j \wedge Ifact(f)^t \wedge \\ \neg(Den_{c,b}(Obl_{b,c}(l \prec f \prec d)^i)^u \wedge u < t) \rightarrow Fulf(Obl_{b,c}(l \prec f \prec d)^i)^t \tag{6.24}$$

$$DViol(Obl_{b,c}(l \prec f \prec d)^i)^j \wedge Den_{c,b}(Obl_{b,c}(l \prec f \prec d)^i)^u \wedge \\ \neg(Ifact(f)^t \wedge t < u) \rightarrow Viol(Obl_{b,c}(l \prec f \prec d)^i)^u \tag{6.25}$$

These rules take into account the timestamps associated with each institutional reality element in order to add other elements with accurate timestamps. The use of relational conditions to assess the temporal ordering of events ends up having a closer reading to the LTL *before* operator.

6.4 Decision-making on Directed Deadline Obligations

The authorization approach described in this chapter enriches the decision-making space of agents concerning norms. Since obligations can be violated, agents (as human delegates) may decide whether to fulfill them or not. Furthermore, because the violation state is determined by the counterparty's choice to denounce this situation, both parties associated with a directed deadline obligation are in a position to decide over it after the deadline. In this section we make a preliminary study on the courses of actions agents may take.

In order to model this decision making process, we need to assess each agent's valuations on the obligation states and facts they are able to bring about. We will write $v_a(f)$ and $v_a(S)$ to denote the valuation agent a makes of fact f or state S, respectively (similarly to the valuation model used by Desai *et al.* [58]). When valuating an obligation's state (namely a fulfillment or a violation), agents should take into account two different sorts of effects. First, since an obligation is taken to be a part of a wider contract that should benefit all participants, the obligation cannot be taken in isolation, as its fulfillment or violation may trigger further commitments. Second, an agent's reputation is affected by whether or not he stands for his commitments. In the following we assume that an agent is capable of anticipating and evaluating the consequences of his actions within a contract.

For an obligation $Obl_{b,c}(f \prec d)$ we have the following valuation constraints for b:

$$
\begin{aligned}
v_b(Obl_{b,c}(f \prec d)) < 0 &: \text{an obligation is a burden to its bearer}\\
v_b(f) < v_b(Obl_{b,c}(f \prec d)) &: \text{there is a heavier cost associated with bringing}\\
&\ \text{about } f\\
v_b(Fulf(Obl_{b,c}(f \prec d))) > 0 &: b \text{ gains from fulfilling his obligation}\\
v_b(Viol(Obl_{b,c}(f \prec d))) < 0 &: b \text{ loses from violating his obligation}
\end{aligned}
$$

The notions of gain and loss for the bearer extend to outside this obligation. For instance, fulfilling an obligation may bring an entitlement (a new obligation where the bearer becomes the counterparty). Violating an obligation will potentially bring penalties to the bearer, hence the negative valuation. In both cases, the reputation of agent b is affected (positively or negatively). Unlike in [58], we do not impose that $v_b(Viol(Obl_{b,c}(f \prec d))) < v_b(f) + v_b(Fulf(Obl_{b,c}(f \prec d)))$. This states that an agent is always better off fulfilling an obligation (taking into account the cost of the fact to bring about) than violating it. However, an agent may be able to exploit a contract flaw by considering that in a specific situation he is better off violating his obligation. Of course that even if the above condition holds, agent b may still choose to violate his obligations, because of other conflicting goals: he may lose with respect to the outcome of this specific contract, but may possibly win across several other contracts.

As for the counterparty c, we have:

$$
\begin{aligned}
v_c(Obl_{b,c}(f \prec d)) > 0 &: \text{an obligation is an asset for the counterparty}\\
v_c(f) > v_c(Obl_{b,c}(f \prec d)) &: c \text{ benefits from } f\\
v_c(Fulf(Obl_{b,c}(f \prec d))) \leq 0 &: c \text{ may acquire obligations after fulfillment}\\
v_c(Viol(Obl_{b,c}(f \prec d))) \geq 0 &: c \text{ may obtain compensations after violation}
\end{aligned}
$$

Note that both fulfillments and violations may bring no value if they have no further consequences in the contract.

In a rough attempt to model the decision making process of a counterparty of an obligation whose deadline was violated, we could state that he should denounce (and thus obtain the obligation's violation) if[4]

$$v_c(f) + v_c(Fulf(Obl_{b,c}(f \prec d))) < v_c(Viol(Obl_{b,c}(f \prec d))).$$

We consider that valuations may possibly vary over time. Were that not the case, the above condition would only need to be checked right after d, at which point the counterparty would either denounce or decide to wait indefinitely for the bearer to fulfill his obligation. For instance, we believe that it makes sense to think of $v_c(f)$ as possibly decreasing with time (like a resource that should be available but is not yet). Even when the above condition does not hold, the counterparty may still prefer to tolerate the less preferred situation of failure for matters of conflicting goals (just as with the bearer).

Until now we have discussed the possibility of agents (both bearers and counterparties) deciding on breach over compliance (either by assessing intra-contract

[4] We assume there is no cost associated with the denouncing action.

consequences or by inter-contract conflicts). But in scenarios enriched with social features agents can exploit, it may be the case that agents decide to behave cooperatively even when they have to bear a contained disadvantage. In such settings, more than being altruistic, agents may try to enhance their trust awareness in the community, from which they will benefit in future interactions or contracts.

6.5 Discussion

The approach to model contractual obligations as directed deadline obligations (and possibly with livelines) has been published in [143] and [146]. A more practical perspective, including implementation, can be found in [144] and [147].

6.5.1 Enacting business contracts

In B2B relationships contracts specify, through obligations, the interdependencies between different partners, and provide legal options to which parties can resort in case of conflict. However, when this joint activity aims at pursuing a common goal, the successful performance of business benefits all involved parties. Therefore, when developing automated monitoring tools, one should take into account that agents may be cooperative enough to allow counterparties' deviations.

Using flexible livelines and deadlines ensures a degree of freedom for agents to make decisions in the execution phase of contracts, which is important for dealing with business uncertainty. Our approach is based on real-world evidence from business contracts (namely the CISG convention [217]), which denotes a flexible and even cooperative facet of trade contracts.

The CISG convention has been widely studied in the research domain of Artificial Intelligence and Law. Moreover, the Legal Expert System project [240] aimed at creating a deductive legal knowledge base, and was applied to CISG. The idea was to deduce appropriate answers to questions about legal states of affairs at any time point, as a result of applying CISG provisions to a concrete case.

6.5.2 Directedness and temporal flexibility

We have developed a novel model for contractual obligations, where these are seen as either *directed deadline obligations* or *directed obligations with liveline and deadline*. Following a claimant theory approach [212], the directed aspect concerns the need to identify the agent who will be authorized to react in case of non-fulfillment. We started from previous theoretical approaches to model such authorizations, and developed a more concrete formalization by linking authorizations with a flexible model of livelines and deadlines. Obligation violations are now dependent on the counterparty motivation to claim them. We have also considered in our model smoother authorizations.

Most implementations of norms in multi-agent systems ignore the need for having directed obligations from bearers to counterparties. The most likely reason for this is that in those approaches obligations are seen as (implicitly) directed from an agent to the normative system itself. It is up to the system (e.g. an electronic organization [228] or an electronic institution [71]) to detect violations and to enforce the norms which are designed into the environment (in some cases they are even regimented in such a way that violation is not possible). On the contrary, our flexible approach towards an Electronic Institution allows agents to specify the norms that will regulate their mutual commitments.

The need to identify two opposite roles in deontic operators is not exclusive of obligations. In [212] the concept of directed permission is described on the basis of *interference* and *counter-performance*. If a party is permitted by another to bring about some fact, the latter is not allowed to interfere with the attempt of the former to achieve that fact. The authors also sustain a relation between directed obligations and directed permissions: $Obl_{b,c}(f) \rightarrow Perm_{b,c}(f)$, that is, if an agent b has an obligation towards an agent c, then b is permitted (by c) to bring about the obliged fact and c is not permitted to interfere. This is very important in international trade transactions, especially when storage costs can be high. Some evidence from CISG [217] brings us once more the same insight:

> *A*rticle 53: The buyer must pay the price for the goods and take delivery of them [...]
>
> *A*rticle 60: The buyer's obligation to take delivery consists: (a) in doing all the acts which could reasonably be expected of him in order to enable the seller to make delivery; and (b) in taking over the goods.

In this case the permission is described in terms of an obligation of the counterparty (the buyer).

Our model of directed obligations with livelines and deadlines has some connections with research on real-time systems, where a time-value function valuates a task execution outcome depending on the time when it is obtained. *Soft* real-time systems use soft deadlines: obtaining the result after the deadline has a lower utility. In contrast, for *hard* real-time systems the deadline is crisp: after it, the result has no utility at all, and missing the deadline can have serious consequences. Our approach seems to be soft with a hard-deadline discretionally declared by the counterparty of the task to achieve. Deadline goals are also analyzed in [105] in the context of goal-directed and decision-theoretic planning. Goals are given a temporal extent and can be partially satisfied according to this temporal component. The authors propose a *horizon* time point somewhere after the deadline, after which there will be no benefit in achieving the goal. In our case the horizon is not static, but can be defined by the counterparty.

6.5.3 Obligation lifecycles

Other authors have proposed different lifecycles for commitments and deontic operators. Directed social commitments are modeled by Pasquier *et al.* [175] in the context of dialogical frameworks. Violated commitments resort to their cancellation, which may bring sanctions. Furthermore, the bearer may explicitly cancel his commitment, allowing the counterparty to apply sanctions. Updating is also allowed through cancellation of the commitment and creation of a new one. A more compact model is that of Wan and Singh in [230], also considering the possibility to update commitments. However, fulfillment and violation are not dealt with explicitly in this model; instead, a commitment is discharged when fulfilled, or else may be canceled. Our approach is more expressive and provides greater flexibility by identifying states that are based on deadlines (and livelines) on which obligations are due. These states, on which counterparties may act, accommodate the possibility to define sanctions with different strengths, making it possible to react to delays differently than violations.

Fornara and Colombetti [78] propose that a commitment's *violated* state can be recovered by including a description of actions to be performed by the debtor (i.e. bearer) in case of violation (which are termed *d-sanctions*). This is seen as an opportunity for the debtor to remedy the violation, and if so the commitment will progress to an *extinguished* state. In case these sanctions fail to produce the desired effect (that is, the debtor still does not conform), the commitment becomes *irrecoverable*, and a second type of sanctions is available (*e-sanctions*) which concerns actions that the norm enforcer is authorized to perform. It seems that with this approach more than one violation is associated with a commitment's lifecycle: failing to comply with the commitment and failing to execute the remedying actions. We prefer to model these as separate obligations (in which the second is a sanction prescribed by a different norm), with which we obtain a more flexible approach that allows us, for instance, to prescribe more than one sanction in a specific violation case.

Taking a cooperative approach to contract fulfillment, the obligation lifecycle model proposed by Sallé [196] includes states that are used in a contract fulfillment protocol. Agents communicate about their intentions to comply with obligations, and in this sense an obligation can be refused or accepted. After being accepted, the obligation may be canceled or complied with. These states are reached according to the enactment of a contractual relationship. Our model, in practice, also requires that agents communicate their intentions regarding an obligation with a violated deadline. In fact, CISG's Article 48 seems to go in this direction, in order to protect the bearer's efforts toward a late fulfillment of the obligation.

6.5.4 Decision-making

The decision-making space of agents concerning contractual obligations is enriched by our model of authorizations. Both parties involved in a directed deadline obligation may have a say regarding its violation. When considering obligations as interlinked through norms in a contract, agents should evaluate the consequences of fulfillment and violation states as prescribed in the contract. Furthermore, in "socially rich" environments, agents should explore the value of future relationships by enhancing their perceived trustworthiness and predisposition to facilitate compliance, something that is made possible by our directed deadline obligations approach.

A model for commitment valuations, on which we have based our decision-making prospect, has been proposed in [58]. However, while their work is centered on checking correctness of contracts, we focus on valuations in the course of a contract execution. We do not assume that a contract is correct from a fairness point of view. This difference in concerns has brought divergent considerations when valuating fulfillment and violation states.

Other authors have studied agent decision-making regarding norm compliance. For instance, violation games, put in perspective of a game-theoretic approach to normative multi-agent systems in [22], model the interactions between an agent and the normative system that is responsible to detect violations and sanction them accordingly. That line of research analyses how an agent can violate obligations without being sanctioned. In our case, while we assume that temporal violations are always detected by the normative environment, we explore agents' decision-making from the point of view of both the bearer and the counterparty of a directed obligation.

Chapter 7

Putting the Normative Environment into Practice

In this chapter, our intention of providing an implementation of an institutional normative environment is satisfied. Based on the proposals we have made in the previous chapters, here we provide a computational infrastructure that may be of assistance both in e-contract establishment and enactment, by providing an automatic monitoring service.

In the next section we will provide an overview of our institutional normative environment implementation. Later on this chapter we will illustrate the use of the normative environment with some typified scenarios. We will show the creation of e-contracts using appropriate representation formalisms (both for contextual information and norms). We will also show possible situations for e-contract enactment, together with the normative environment's response as an e-contract monitoring service.

7.1 A Rule-based Implementation

Our approach towards implementing a normative environment is based on a declarative representation of rules and norms, together with the normative state. As it follows from Chapter 5, rules and norms lend themselves to a rule-based implementation. Furthermore, since the normative environment is based on the occurrence of events, using a data-driven (i.e., forward chaining) rule engine is appropriate. This kind of inference engine was initially known as a production system, widely used in the realm of expert systems [90]. One of the main characteristics of production systems is the conceptual separation between knowledge representation and its application – in our case, between norms and their use in contractual relationships, through monitoring rules.

In general terms, a rule-based system is composed of three main components:

a (i) *knowledge base* contains a collection of rules that represent the problem-solving abilities to be applied in a specific problem domain; (ii) *working memory* includes the available data that characterize the state of a problem as it is being solved, including partial results; an (iii) *inference engine* tries to match rules in the knowledge base with data in working memory and decides, using appropriate conflict resolution strategies, which rules to apply in each step of the reasoning process. In our case, rules (both constitutive and institutional) and norms compose the knowledge base, while the normative state is included in working memory. The inference engine, besides applying rules that respond to events as they are added to working memory, will be responsible to put in action the norm activation defeasibility model described in Section 5.6.

From the available choices, we selected Jess [82] as our rule-based engine. Jess includes a number of interesting features that we exploit in our implementation, namely:

- a very efficient inference-engine using an enhanced version of the Rete algorithm, which optimizes the pattern matching phase of the inference process;
- an organization of rules and templates in modules, which allows us to organize the complexity of having multiple norms that apply to different contexts with hierarchical relationships;
- a representation of working memory elements supported by slot-based template declarations, also enabling slot inheritance through template extension;
- an easy integration with Java, by enabling the use of Java classes from Jess as a scripting language and by allowing a full Java application to embed a Jess component.

Although the first three items listed above are advantageous in the process of implementation of a normative environment, the last item should not be disregarded, since it is also our concern to integrate the normative environment within a full-fledged Electronic Institution Platform, which is being developed at LIACC-NIAD&R as a JADE [18] agent-based system.

While explaining our implementation throughout the following subsections, we will intentionally hide some intricate details that are more difficult to grasp for a non Jess-familiar reader. Nevertheless, in Appendix A we provide a full Jess code listing of our approach.

7.1.1 Contexts

We will start by providing a means to represent contexts and their hierarchical relations. Each e-contract that is created will give birth to a context inside the normative environment.

A Jess template (defined with the `deftemplate` construct) aggregates the basic information for a context definition, including an id, the name of its parent (super) context, a starting date for the associated contract and a list of participants:

```
(deftemplate MAIN::context
    "Context for norms"
    (slot id) (slot super-context (default INSTITUTIONAL-NORMS))
    (slot when) (multislot who) )
```

Initially, a top level institutional context is available, from which every context descends. It is represented by identifier `INSTITUTIONAL-NORMS`, which is the default value for the `super-context` slot.

The contextual information that is defined for each specific contract is based on the following template:

```
(deftemplate MAIN::contextual-info
    "Contextual information"
    (slot context) )
```

The typification of contexts is made possible by extending both the `context` and `contextual-info` templates. We will provide some examples in Section 7.2.

When creating a new context, a new Jess module is defined (by using the `defmodule` construct) within which specific norms may be included. This definition of norms inside modules is useful for handling the complexity associated with having a potentially large number of norms defined in different contexts[1]. Again, in Section 7.2 we provide example uses of context creation and context-specific norm definitions.

7.1.2 Normative state

In order to properly represent institutional reality elements that will compose the normative state, we define a set of templates and take advantage of Jess template inheritance, via the `extends` keyword.

```
(deftemplate MAIN::IRE
    "Institutional reality element"
    (slot context) (slot when) )

(deftemplate MAIN::ifact extends IRE
    "Institutional fact"
    (multislot fact) )
```

[1]While this Jess feature may also be used to control rule firing, we use it merely with an organizational aim.

```
(deftemplate MAIN::time extends IRE
    "Time event" )

(deftemplate MAIN::obligation extends IRE
    "Obligation"
    (slot bearer) (slot counterparty) (multislot fact)
    (slot liveline (default 0)) (slot deadline) )

(deftemplate MAIN::liveline-violation extends IRE
    "Liveline violation of an obligation"
    (slot obl) (slot ifa) )

(deftemplate MAIN::deadline-violation extends IRE
    "Deadline violation of an obligation"
    (slot obl) )

(deftemplate MAIN::fulfillment extends IRE
    "Obligation fulfillment"
    (slot obl) (slot ifa) )

(deftemplate MAIN::violation extends IRE
    "Obligation violation"
    (slot obl) )

(deftemplate MAIN::denounce extends IRE
    "Temporal violation denouncement"
    (slot obl) )
```

Deadline obligations can be modeled by letting the liveline slot of obligations get its default value of `0`, which means no liveline (it can never be the case that the obliged fact is obtained before the liveline). In liveline-violations and fulfillments we included, besides the underlying obligation, a reference to the institutional fact (`ifa`) based on which these elements were obtained. The assertion of time events (e.g. associated with livelines and deadlines) that are relevant in a specific context is made by scheduling time alerts using a system clock. Inside `ifact` and `obligation`, the unstructured content of a `fact` (which is declared as a `multislot`, that is, a list of values) is meant to increase readability in the examples provided throughout this chapter.

For practical reasons we also add two events that allow us to grasp the lifetime of a context:

```
(deftemplate MAIN::start-context extends IRE
    "To indicate that a context has started" )

(deftemplate MAIN::end-context extends IRE
    "To indicate that a context has ended" )
```

These institutional reality elements (whose instances will be part of the normative state) provide a temporal window that allows us to filter out any events related to contexts that are no longer active. We establish this connection between an *active context* and the `start-context` and `end-context` events by defining an additional template and a Jess rule (through the `defrule` construct):

```
(deftemplate MAIN::active-context
    "To indicate that a context is active"
    (slot context) )

(defrule MAIN::active-context-rule
    "Active context state"
    (logical (start-context (context ?ctx)))
    (logical (not (end-context (context ?ctx))))
    =>
    (assert (active-context (context ?ctx))) )
```

This rule, in particular, states that while a context is started and not ended, it is active. The `logical` keyword produces a dependence between the rule's conditions and the rule's assertion (i.e., the active context state). Therefore, if the context has not yet started or if it has ended, the `active-context` Jess fact will not be part of working memory. We will make use of this context property in constitutive rules.

Jess rules are quite simple to read. The left-hand-side (before the `=>`) is composed of Jess fact patterns, which may be bound to variables (identifiers starting with a question mark '?') to be used later. In the right-hand-side we may include any function calls, which in our case consist of assertions to be made in the normative state.

7.1.3 Brute facts, roles and constitutive rules

Brute facts are encoded as illocutions in the normarive environment:

```
(deftemplate MAIN::bfact
    "Brute fact"
    (slot agent) (slot context) (multislot statement) (slot when) )
```

In this case we are making more evident a reference to the `context` regarding which a statement is being made that something has happened at instant `when`.

Role enacting agents are represented by an appropriate template:

```
(deftemplate MAIN::rea
    "Role enacting agent - agent takes a role"
    (slot agent) (slot role) )
```

The example constitutive rules shown in Section 5.4.2 have a straightforward translation to Jess rules (some are included in Appendix A). For instance:

```
(defrule CONSTITUTIVE-RULES::payments
    "Constitutive rule for acknowledging payments"
    (bfact (agent ?a) (context ?ctx) (when ?wh)
        (statement currency-transfered $?data) )
    (active-context (context ?ctx))
    (rea (agent ?a) (role Bank))
    =>
    (assert (ifact (context ?ctx) (when ?wh)
                (fact payment $?data) ) ) )
```

This constitutive rule is based on a brute fact, and checks that the statement is made regarding an active context. A constitutive rule that allows us to create a denounce from the corresponding institutional fact is as follows:

```
(defrule CONSTITUTIVE-RULES::denounces
    "Acknowledging denounces"
    (ifact (context ?ctx) (when ?wh)
        (fact msg-delivery ref ? from ?fr to ?to msg denounce $?fact) )
    ?obl <- (obligation (context ?ctx) (bearer ?to) (counterparty ?fr)
                (fact $?fact) )
    =>
    (assert (denounce (context ?ctx) (obl ?obl) (when ?wh))) )
```

In this case we search for an obligation whose fact to bring about matches the denounced fact. We bind such obligation to a variable (`?obl`) and use it as a reference inside the `denounce` element.

7.1.4 Institutional rules: monitoring contractual obligations

Institutional rules for monitoring contractual obligations (see Section 6.3.2), are translated into Jess rules. For instance:

```
(defrule INSTITUTIONAL-RULES::detect-liveline-violation
    "Detect a livelive violation of an obligation"
    ?obl <- (obligation (context ?c) (fact $?f) (liveline ?liveline))
    ?ifa <- (ifact (context ?c) (fact $?f) {when < ?liveline})
    =>
    (assert (liveline-violation (context ?c) (when ?ifa.when)
                (obl ?obl) (ifa ?ifa) ) ) )
```

In this case, the rule is applied when there is an obligation and an institutional fact with the same fact contents, and it binds these two elements with variables. If the institutional fact occurs before the liveline, a liveline violation is asserted, passing to it references to the elements used to fire the rule.

The whole set of monitoring rules is included in Appendix A.

7.1.5 Norms and defeasibility

At a first glance, norms are represented in Jess quite similarly to institutional rules. They use patterns in their left-hand-side and they assert obligations (or an `end-context` element signaling a contract termination) in their right-hand-side. And in fact for contract-specific user-defined norms this is all the user needs to know. The concrete Jess coding of norms in our normative environment is, however, a bit more intricate, as we now explain.

The conflict resolution strategies set forth by Jess do not distinguish rules (norms) according to the normative contexts in which they are defined (see Section 5.6.2). Although it is possible to program a specific resolution strategy, our norm defeasibility model demands that not only we decide which norm to apply, but also that we prevent defeated norms from being fired. Since our normative state is monotonic (i.e. we never retract institutional reality elements), we need to guide the inference engine in order to apply norms in a selective way.

Our approach consists of a special handling process of institutional reality elements and on some additions made to the representation of norms in Jess. Each IRE includes an additional slot indicating its *processing context*. When an IRE is first added to the normative system, this `processing-context` slot is set to be the context that the IRE refers to. After norms that are defined in that context have had a chance to fire, the IRE's processing context is set to be the `super-context` of the current processing context. This step is repeated until we cross the `INSTITUTIONAL-NORMS` context, which is the top normative context. In Appendix A we include an implementation of this algorithm using Jess rules.

Norms will be allowed to fire using a set of IREs if and only if at least one of them has the norm context as its processing context. Furthermore, while firing norms we make sure that the normative environment remembers what normative state fractions have been used to fire norms throughout contexts. A norm will only fire if there was no other norm, defined at a different context, that fired on the same set of IREs. If there was such a norm, then it necessarily is defined in a sub-context, because of the way the `processing-context` slot is updated.

The general simplified appearance of a norm is as illustrated in Figure 7.1. Each norm is defined at a specific module (`INSTITUTIONAL-NORMS` or any context module), which is expressed by the label between `defrule` and the `::` separator. In each left-hand-side pattern any templates extending those presented (`context`, `contextual-info` or `IRE`) may be used.

In order to implement the norm defeasibility model as explained above, a

```
(defrule <NORM-CONTEXT>::simplified-norm-codification
    "Brief description of the norm"
    (context (id ?ctx) ...)
    (contextual-info (context ?ctx) ...)
    ...
    (IRE (context ?ctx) ...)
    ...
    =>
    (assert (obligation (context ?ctx) ...))
    ... )
```

Figure 7.1: Simplified norm codification.

```
(defrule <NORM-CONTEXT>::concrete-norm-codification
    "Brief description of the norm"
    (context (id ?ctx) ...)
    (contextual-info (context ?ctx) ...)
    ...
    ?ire_1 <- (IRE (context ?ctx) ... (processing-context ?pc_1))
    ...
    ?ire_n <- (IRE (context ?ctx) ... (processing-context ?pc_n))
    ; at least 1 IRE has this module as its processing-context?
    (test (member$ <NORM-CONTEXT> (list ?pc_1 ... ?pc_n)))
    ; is there no norm-fired-on with these IREs and another context?
    (not (norm-fired-on
             (ires $?ires&:(equal-sets? ?ires (list ?ire_1 ... ?ire_n)))
             (context ~<NORM-CONTEXT>)) )
    =>
    (assert (obligation (context ?ctx) ...))
    ...
    ; save norm-fired-on information
    (assert (norm-fired-on (ires ?ire_1 ... ?ire_n)
                (context <NORM-CONTEXT>) ) ) )
```

Figure 7.2: Concrete norm codification.

norm is in fact coded in the normative environment as shown in Figure 7.2. The template `norm-fired-on` is used to save information regarding contexts in which norms have fired on particular IREs. The use of a tilde (~) in the `context` slot of pattern `norm-fired-on` negates the value following it: in this case, we want to eliminate the possibility of having a norm fired on the same IREs but that is defined in a *different* context. The tests that are made can be simplified in case the norm includes only one IRE pattern.

7.2 E-Contracting Scenarios

E-contracts are established as a result of negotiation among autonomous agents. Nevertheless, the normative environment does not require e-contracts to be negotiated within the Electronic Institution Platform. In order to be monitorable by the normative environment, however, e-contracts should be written according to the XML Schema specification provided in Appendix E. This schema enables agents to exploit the adaptability and extensibility features of the normative framework. Besides including a header section where information regarding context definition is to be provided, the schema also allows norms to be specified. When handed to the normative environment, an XML contract representation based on this schema is converted into Jess constructs, giving birth to a new context that includes contextual information as defined in the contract header and, optionally, context-specific norms.

The specification of types of contracts, together with their default normative core, is a knowledge engineering effort that should take place so that the normative environment can be exploited to its full extent. Legislative bodies or business practices are important sources for carrying out this task. While being less ambitious in our quest, in this section we will illustrate the use of the normative environment using some e-contracting examples. We will not express contracts using XML, but will instead concentrate on the e-contract representation as used inside the normative environment. In order to increase readability, we will show the simplified version of norms, as explained in the previous section.

7.2.1 Contract of sale

A contract of sale is a contract between two parties by which a seller agrees to sell to a buyer a specific asset for a given amount of money. Although we have pointed out, in Section 4.2, that this kind of contract is perhaps too simplistic to demand for electronic monitoring tools, we will use it as a starting example in the task of exploiting our normative environment.

Appendix B includes Jess code for defining a contract of sale:

- the `contract-of-sale` template construct as an extension of the `contract` template;
- the `contract-of-sale-data` contextual information as an extension of the `contextual-info` template;
- a set of predefined institutional norms regulating contracts of sale.

The first couple of norms define the normal situation: when the contract starts, the seller will be obliged to deliver the product (norm `CoS_delivery`); when he does so, the buyer will be obliged to make the corresponding payment (norm

CoS_payment). The two norms following these prescribe what should happen in abnormal situations: norm CoS_no-delivery dictates that in case the seller violates his obligation to deliver the product then the contract shall be terminated; on the other hand, norm CoS_no-payment indicates that any failure to meet the payment deadline shall result in an additional obligation to pay an interest of 10% applied on the amount due. Finally, norm CoS_end-of-contract captures the normal situation for reaching the end of a contract execution: both delivery and payments have occurred, and there are no pending (i.e., not fulfilled) payment obligations (e.g. interests).

The two sanctioning norms are applied in two different types of violations: while the interest to be applied on the payment due is automatic once a deadline violation is detected, the prescription of a contract cancellation by norm CoS_no-delivery requires the buyer to denounce the inability of the seller to fulfill the delivery. It is therefore up to the buyer to wait further and accept a delayed delivery or not. If the agreed upon contract conditions are important enough, allowing a counterparty deviation (and hence taking a cooperative attitude regarding the compliance of the contract) may be a good decision.

A typical contract of sale

According to the definitions presented above, a new contract of sale may be created just by providing the corresponding template instantiations. For instance, inside the normative environment we may have:

```
(contract-of-sale
    (id CoS-T12)
    (when ...)
    (who ForOffice LIACC) )

(contract-of-sale-data
    (context CoS-T12)
    (seller ForOffice) (buyer LIACC)
    (product computer-desk) (quantity 10) (unit-price 20.0)
    (delivery-rel-liveline 3000) )
```

In this contract LIACC is purchasing computer desks to ForOffice, and does not want them before the stipulated liveline. We will now show some possible outcomes of the monitoring process in different contract enactment situations.

In Appendix C, figures C.1-C.5 show the response of the normative environment to different enactment situations, from the start till the end of contract execution. The listings in these figures include relevant *IRE* that are produced by rules and norms, together with institutional facts originated from agent actions (these are marked with an asterisk). Relevant time events (associated with livelines and deadlines) triggered by a system clock are also shown.

Figure C.1 shows the normative state after a perfect contract enactment, where everything goes as agreed. No temporal violations are detected in this case, since agents abide to their obligations.

Figures C.2-C.4 depict different enactment outcomes in which delivery problems are detected. In enactments at Figures C.2 and C.3 the delivery liveline or deadline is violated (and detected by rules adding `f-40` and `f-43`, respectively), while the counterparty (`LIACC`) does not denounce this situation. Enactment at Figure C.4 shows a situation in which the counterparty chooses to denounce (`f-41`) the violation of the delivery deadline (as detected in `f-39`); as prescribed by contractual norm `CoS_no-delivery`, this results in a contract end (`f-45`). Finally, Figure C.5 shows an enactment in which the payment deadline was violated, causing an interest to be applied according to contractual norm `CoS_no-payment`. Agent `LIACC` eventually payed both the price (`f-57`) and the interest (`f-60`), bringing the contract to an end (`f-62`), according to norm `CoS_end-of-contract`.

An atypical[2] contract of sale

So far we have not made any use of the flexibility of the normative framework provided by the norm defeasibility model. In this section we start doing so.

The way we defined templates that contain contract of sale data allows us to adjust some parameters simply by overriding default values (in fact in the previous contract example we did so for the `delivery-rel-liveline` slot). By exploiting the defeasibility feature of the normative framework, we may also create a contract of sale that does not work exactly as prescribed by predefined institutional norms.

As an example, consider the following contract:

```
(contract-of-sale
    (id CoS-A12)
    (when ...)
    (who CPUStore LIACC) )

(contract-of-sale-data
    (context CoS-A12)
    (seller CPUStore) (buyer LIACC)
    (product laptop-computer) (quantity 1) (unit-price 800.0) )

(defrule CoS-A12::void
    "Nothing after delivery"
    (contract-of-sale (id CoS-A12))
    (fulfillment (context CoS-A12) (obl ?obl) (when ?w))
    ?obl <- (obligation (context CoS-A12) (bearer ?s) (counterparty ?b)
                (fact delivery $?))
    => )
```

[2]Atypical here means not exactly as predefined in the normative environment. It is not meant to indicate that the corresponding contractual norms are not common in the real world.

```
(defrule CoS-A12::payment
    "After delivery and no return within period, buyer obligation to pay"
    (contract-of-sale (id CoS-A12))
    (contract-of-sale-data (context CoS-A12) (seller ?s) (buyer ?b)
        (product ?p) (quantity ?q) (unit-price ?up)
        (payment-rel-deadline ?prd) )
    (fulfillment (context CoS-A12) (obl ?obl) (when ?w))
    ?obl <- (obligation (context CoS-A12) (bearer ?s) (counterparty ?b)
                (fact delivery $?) )
    (time (context CoS-A12) (when =(+ ?w 10000)))
    (not (ifact (context CoS-A12)
             (fact delivery ref ? from ?b to ?s product ?p quantity ?q)
             {when <= (+ ?w 10000)} ) )
    =>
    (assert (obligation (context CoS-A12) (bearer ?b) (counterparty ?s)
                (deadline (+ ?w ?prd))
                (fact payment ref CoS-A12 from ?b to ?s
                              amount (* ?q ?up)) ) ) )

(defrule CoS-A12::end-of-contract
    "Delivery and product return lead to end of contract"
    (contract-of-sale (id CoS-A12))
    (contract-of-sale-data (context CoS-A12) (seller ?s) (buyer ?b)
        (product ?p) (quantity ?q) (unit-price ?up)
        (payment-rel-deadline ?prd) )
    (fulfillment (context CoS-A12) (obl ?obl_del))
    ?obl_del <- (obligation (context CoS-A12) (bearer ?s) (counterparty ?b)
                    (fact delivery ref ? from ?s to ?b
                                   product ?p quantity ?q) )
    (ifact (context ?CoS-A12)
           (fact delivery ref ? from ?b to ?s product ?p quantity ?q)
           (when ?w_ret) )
    =>
    (assert (end-context (context CoS-A12) (when ?w_ret))) )
```

In this contract `LIACC` is purchasing a laptop to `CPUStore`. This contract includes three specific norms, where the parties specify that a trial period is set within which the laptop may be returned to the seller. Norm `CoS-A12::void` is meant to defeat norm `INSTITUTIONAL-NORMS::CoS_payment` (according to the defeasibility model), since we do not want to raise the payment obligation as soon as LIACC gets the laptop. Norm `CoS-A12::payment` establishes that the payment obligation arises only if the laptop is not returned within a specific time period. Finally, norm `CoS-A12::end-of-contract` states that the contract is considered terminated upon a product return. The rest of the norms defined at the `INSTITUTIONAL-NORMS` level remain applicable to this contract. In Appendix C,

Figure C.6 shows a smooth enactment of this contract, while Figure C.7 depicts a situation where `LIACC` has decided to return the laptop.

7.2.2 Standing agreements

In the previous section we have used a one-off business case, based on the contract of sale. However, it is with standing agreements that we make most sense out of the hierarchical normative structure that we have included in our normative environment design. In this section we try to provide an illustrating scenario.

As we have noted in Section 4.2, in many cases a contractual relation forms the business context for further contracts. Therefore, it is within some previously established cooperation agreement that more specific and short-term contracts are formed. We now build a scenario with three hierarchical levels.

A group of agents (representing companies) forms a cluster that agrees on a set of principles regarding potential collaboration in business opportunities. Each agent has competences in a specific area (e.g. as defined in ISIC[3]). This acquaintance forum is represented, under our approach, by the following contract type and contextual information:

```
(deftemplate MAIN::cluster extends contract
    "A cluster of acquainted enterprises" )

(deftemplate MAIN::acquainted-info extends contextual-info
    "International Standard Industrial Classification for each agent"
    (slot agent)
    (slot isic-code) )
```

An example of a norm governing a cluster is the following:

```
(defrule INSTITUTIONAL-NORMS::cluster_cfp-prop
    "A cfp demands for a proposal within a reasonable time"
    (cluster (id ?ctx))
    (acquainted-info (context ?ctx) (agent ?ag1) (isic-code ?isic))
    ?ifa <- (ifact (context ?ctx) (when ?w)
                (fact cfp ref ?ref from ?ag to ?ag1
                          isic-code ?isic $?))
    =>
    (assert (obligation (context ?ctx) (bearer ?ag1) (counterparty ?ag)
                (deadline (+ ?w 10000))
                (fact proposal ref ?ref from ?ag1 to ?ag
                               isic-code ?isic $?) ) ) )
```

[3]International Standard Industrial Classification of All Economic Activities, Rev.4 – `http://unstats.un.org/unsd/cr/registry/isic-4.asp`

This norm simply states that members of a cluster are expected to answer call-for-proposals related with their ISIC code within a reasonable time. While we will not specify any consequence in case of violation, this norm can be seen as a behavior guideline to which deviations could be handled in the future.

A specific cluster

With the following illustration we show the potential of the hierarchical normative structure, by creating a cluster that includes specific particularities. These are expressed as norms to be applied to "cooperation agreements" that might be established among a subset of the cluster's members. Therefore, norms created at run-time (i.e., they are not defined at the top institutional level) will regulate e-contracts that are to be created at a latter stage.

We now create cluster `Cluster-12345`. First, the cluster's definitions:

```
(cluster
    (id Cluster-12345)
    (when ...)
    (who E1 E2 E3 E4 E5) )

(acquainted-info
    (context Cluster-12345) (agent E1) (isic-code 261) )

(acquainted-info
    (context Cluster-12345) (agent E2) (isic-code 261) )

(acquainted-info
    (context Cluster-12345) (agent E3) (isic-code 262) )

(acquainted-info
    (context Cluster-12345) (agent E4) (isic-code 262) )

(acquainted-info
    (context Cluster-12345) (agent E5) (isic-code 263) )
```

We thus have a cluster composed of five different agents, although some have the same ISIC code.

Then, the notion of cooperation agreement as defined in `Cluster-12345`:

```
(deftemplate Cluster-12345::coop-agreement extends contract
    "A cooperation agreement where parties promise to supply products
     within agreed criteria" )
```

```
(deftemplate Cluster-12345::coop-effort extends contextual-info
    "Cooperation effort of a coop-agreement partner"
    (slot agent)
    (slot product)
    (slot min-qt (default 1))
    (slot max-qt (default 1))
    (slot unit-price (default 0.0)) )
```

Each `coop-effort` entrance specifies the criteria (namely, the ordered quantity range) according to which a partner commits to sell the product for a given unit price. This is expressed by the following norms governing cooperation agreements that might be created within context `Cluster-12345` (this is the context where the norms are defined):

```
(defrule Cluster-12345::CA_order-accept
    "A partner must accept orders conforming to his coop-effort promise"
    (coop-agreement (id ?ctx) (who $?who))
    (coop-effort (context ?ctx) (agent ?ag1) (product ?p)
        (min-qt ?min_q) (max-qt ?max_q) )
    (ifact (context ?ctx) (when ?w)
           (fact order ref ?ref from ?ag to ?ag1
                       product ?p quantity ?qt) )
    (test (member$ ?ag $?who))
    (test (and (>= ?qt ?min_q) (<= ?qt ?max_q)))
    =>
    (assert (obligation (context ?ctx) (bearer ?ag1) (counterparty ?ag)
                (deadline (+ ?w 5000))
                (fact accept ref ?ref from ?ag1 to ?ag
                             product ?p quantity ?qt) ) ) )

(defrule Cluster-12345::CA_accept-CoS
    "Accepting an order means to establish a new contract-of-sale"
    (coop-agreement (id ?ctx))
    (coop-effort (context ?ctx) (agent ?ag1) (product ?p)
        (unit-price ?upr) )
    (ifact (context ?ctx) (when ?w_order)
        (fact order ref ?ref from ?ag to ?ag1 product ?p quantity ?qt) )
    (ifact (context ?ctx) (when ?w_accept)
        (fact accept ref ?ref from ?ag1 to ?ag product ?p quantity ?qt) )
    =>
    ; establish contract-of-sale
    (create-contract-of-sale super-context ?ctx when ?w_accept
                             seller ?ag1 buyer ?ag product ?p
                             quantity ?qt unit-price ?upr) )
```

Norm `Cluster-12345::CA_order-accept` expresses the obligation of a partner agent to accept orders respecting the promised criteria, provided that the ordering agent is also a cooperation agreement partner. This however does not prevent the agent from accepting orders not matching the quantity criteria. Obliged or not, however, norm `Cluster-12345::CA_accept-CoS` establishes that should the agent accept an order, a contract of sale (see Section 7.2.1) is created that will operationalize the product supply. In this case we extend the definition of norms in order to make use of a Jess function allowing us to create a new contract of sale.

In order to show the use of these norms, we define the following cooperation agreement `CA-123` among three of the cluster's agents:

```
(Cluster-12345::coop-agreement
    (id CA-123)
    (super-context Cluster-12345)
    (when ...)
    (who E1 E3 E5) )

(Cluster-12345::coop-effort
    (context CA-123) (agent E1) (product motherboard)
    (min-qt 90) (max-qt 100) (unit-price 40.0) )

(Cluster-12345::coop-effort
    (context CA-123) (agent E2) (product desktop-computer)
    (min-qt 90) (max-qt 100) (unit-price 500.0) )

(Cluster-12345::coop-effort
    (context CA-123) (agent E2) (product monitor)
    (min-qt 90) (max-qt 100) (unit-price 150.0) )

(Cluster-12345::coop-effort
    (context CA-123) (agent E3) (product modem)
    (min-qt 90) (max-qt 100) (unit-price 120.0))
```

This cooperation agreement has been signed between agents `E1`, `E2` and `E3`. The latter committed to supply two different products, while the others have only one kind of contribution. It is worth pointing out that the super context for `coop-agreement` `CA-123` is `Cluster-12345`: this will enable context `CA-123` to inherit norms defined in context `Cluster-12345` that apply to cooperation agreements.

In Appendix C, Figure C.8 shows what happens when agent `E3` orders 90 motherboards to agent `E1`. When the order is placed (`f-88`) an obligation to accept the order is added (`f-89`), which when fulfilled (`f-93`) causes a contract of sale to be created (`f-95`, `f-96`). What follows is a smooth enactment of the contract of sale, similar to that of Figure C.1.

7.2.3 Workflow contracts

A final example we would like to provide is that of a multi-party contract between a set of companies that agree on establishing some kind of inter-organizational workflow practice among them, as modeled by Xu [239]. According to this approach, two main concepts may be identified: *action* and *constraint*. An action indicates who is responsible for executing a task and towards whom that task is to be performed (these are the *sender* and *receiver* of the action), together with a relative deadline. A constraint expresses restrictions on the occurrence order of actions in a business process. Abstracting from this model, we define a *workflow contract* as a contract where obligations regard actions to be executed in a particular order (i.e., with ordering constraints) between contractual agents.

The workflow contract is represented as follows:

```
(deftemplate MAIN::workflow-contract extends contract
    "A workflow contract" )

(deftemplate MAIN::action extends contextual-info
    "An action to be taken by a workflow-contract participant"
    (slot name)
    (slot sender)
    (slot receiver)
    (slot rel-deadline (default 0)) )

(deftemplate MAIN::constraint extends contextual-info
    "A constraint on the order of two actions"
    (slot action1)
    (slot action2) )
```

Two norms govern workflow-contracts. Norm `WfC-starting-actions` responds to an action starting the workflow process. We identify this action by checking that there is no precedence constraint over it. Once the starting action has been executed, we prescribe as obligations any actions constrained by that action.

```
(defrule INSTITUTIONAL-NORMS::WfC-starting-actions
    "Dealing with the starting action"
    (workflow-contract (id ?ctx))
    (action (name ?action) (sender ?fr) (receiver ?to))
    ; check that this is a starting action
    (not (constraint (action2 ?action)))
    ; get a next-action depending on this one
    (constraint (action1 ?action) (action2 ?next-action))
    (action (name ?next-action) (sender ?s) (receiver ?r)
        (rel-deadline ?d) )
```

```
    ; when the action is executed
    (ifact (context ?ctx) (when ?w)
        (fact ?action ref ?ref from ?fr to ?to) )
    =>
    ; create an obligation for the next-action
    (assert (obligation (context ?ctx) (bearer ?s) (counterparty ?r)
                (deadline (+ ?w ?d))
                (fact ?next-action ref ?ref from ?s to ?r) ) ) )
```

Norm (`WfC-obliged-actions`) handles the rest of the workflow sequence:

```
(defrule INSTITUTIONAL-NORMS::WfC-obliged-actions
    "Dealing with obliged actions"
    (workflow-contract (id ?ctx))
    (action (name ?action) (sender ?fr) (receiver ?to))
    ; get a next-action depending on this one
    (constraint (action1 ?action) (action2 ?next-action))
    (action (name ?next-action) (sender ?s) (receiver ?r)
        (rel-deadline ?d))
    ; when the action is fulfilled
    (fulfillment (context ?ctx) (when ?w) (obl ?obl))
    ?obl <- (obligation (context ?ctx)
                (fact ?action ref ?ref from ?fr to ?to))
    ; check that no precedence is pending for next-action
    (not (and (constraint (action1 ?prev-action) (action2 ?next-action))
              (test (neq ?prev-action ?action))
              (action (name ?prev-action) (sender ?s1) (receiver ?r1))
              (not (and (fulfillment (context ?ctx) (obl ?obl_))
                        ?obl_ <- (obligation (context ?ctx)
                                     (fact ?prev-action ref ?ref
                                           from ?s1 to ?r1) ) ) ) ) )
    =>
    ; create an obligation for the next-action
    (assert (obligation (context ?ctx) (bearer ?s) (counterparty ?r)
                (deadline (+ ?w ?d))
                (fact ?next-action ref ?ref from ?s to ?r) ) ) )
```

When an obliged action is fulfilled, we get a next action waiting for (i.e. constrained by) this one and check if every precedence of that action is fulfilled. If so, we add an obligation concerning that action.

Car insurance scenario

The example provided by Xu [239] regards a car insurance scenario, which we will try to replicate with our approach. A brief explanation is as follows:

> "This case outlines the manner in which a car damage claim is handled by an insurance company (AGFIL). The contract parties work together to provide a service level which facilitates efficient claim settlement. The parties involved are called Europ Assist, Lee Consulting Services, Garages and Assessors. Europ Assist offers a 24-hour emergency call answering service to policyholders. Lee C.S. coordinates and manages the operation of the emergency service on a day-to-day level on behalf of AGFIL. Garages are responsible for car repair. Assessors conduct the physical inspections of damaged vehicles and agree upon repair figures with the garages." (from [239], p. 121)

Appendix D includes the definition of the car insurance scenario, including:

- a `workflow-contract` instance with participating agents: `policyholder`, `agfil`, `euro_assist`, `lee_cs`, `garage` and `assessor`;
- an `action` instance for each action to be executed by an agent;
- a `constraint` instance for each action pair ordering restriction.

Also included in Appendix D is a normal enactment when the policy holder places a phone claim. Every constraint is satisfied as a result of their usage in norms prescribing obligations for the actions to be taken.

7.3 Discussion

The illustration, through practical examples, of the functioning of our normative environment proposal has been our concern mainly in [137], [144] and [147]. We have also made a proposal for an XML contract schema [138] that allows agents to take advantage of the adaptability and extensibility features of the normative framework, by going from a mere predefined contract type instantiation to the inclusion of contract specific norms.

7.3.1 Declarative rule-based approach

Our implementation using a forward-chaining rule-based approach is applicable to run-time monitoring of contracts. A requirement of this kind of usage however is that events are reported in a timely fashion to the normative environment. We assume that agents are interested in publicizing their abidance to commitments.

An advantage of a declarative normative framework is that agents can have access to their applicable norms, and can use what-if analysis to predict and reason about possible outcomes of both their and other agents' actions. This may be important in contracting situations, allowing potential contract breaches to be detected and prevented, or to minimize their damages. For that, the monitoring capabilities of our implementation may be used as a tool to alert agents when certain contract-related events are eminent, such as a forthcoming deadline. Jess allows for an easy integration of our monitoring rules implementation with other rules including function calls that address the level of responsiveness that is intended.

We have shown how the normative environment may effectively monitor contract enactment at run-time. Monitoring rules may also be used *a posteriori*, in order to check off-line if contractual norms were indeed followed by every partner. In this case, after collecting all events concerning a contract, the inference engine may run in order to check if the contract was enacted in a conforming way.

A less demanding approach in terms of run-time state maintenance is that of event-calculus, such as in the usage proposed by Knottenbelt [127]. Events initiate or terminate fluents regarding the normative positions of agents within a contract. Appropriate rules capture the fulfillment or violation of obligations. A meta-interpreter enables an agent to query what obligations are active with respect to a contract. The distinguishing feature in this approach is that the truth value of fluents (e.g. obligation states) is dynamically obtained when asked for, instead of a monitoring process that keeps maintaining the normative state through real-time assertions. However, in our approach we do need to keep an updated normative state in order to provide a run-time contract monitoring facility.

Other approaches have followed a rule-based approach to norm representation and monitoring. For instance, in [87] the authors seem to implement in a backward-chaining logic program (based on Prolog) the semantics of a forward-chaining production system. We follow a more intuitive approach by employing a forward-chaining shell.

7.3.2 Normative settings

Some approaches to define normative languages illustrate their application in simplistic auction scenarios (e.g. [79][87]). With a bit of effort, our approach could also be used to model these cases. However, instead of providing these "pre-formatted institutions", the rationale behind our normative environment development is quite different since the beginning: that of allowing agents to create the normative relationships they want to establish.

In Section 7.2 we have provided some examples of contracts that might be established by agents. We have extended the kinds of normative consequences that norms may bring, by showing, in Section 7.2.2, a case where agent actions within a contract have as a consequence the creation of a new contract. In rigor, this demands for a refinement on our norm definition (which has been initially defined as a rule that prescribes obligations). Another way of formally taking into account this possibility is to read the creation of a new contract as a "counts-as" conditional, e.g. by saying that an order and its acceptation count as the creation of a contract of sale between the involved parties.

A question that arises when we look at contractual responses to violations, which mostly consist on prescribing, as sanctions, new obligations, is what should happen when an agent does not fulfill his contrary-to-duties. Our current implementation does not account for further direct consequences within the normative environment. However, being integrated in an overall Electronic Institution Platform, the normative environment does provide reports on contract enactment to a Computational Trust and Reputation service, as explained in Section 4.1. Therefore, the normative environment does in fact contribute to enable the application of "social sanctions" by the Electronic Institution.

In the next chapter we provide an initial approach towards enriching a normative environment with corrective measures when agents tend not to fulfill their commitments.

Chapter 8

Norm Enforcement

As discussed in Chapter 4, in order to put its regulations into practice an EI needs to carry out two complementary tasks. In the previous chapters we have mainly dealt with the monitoring process. The other task – enforcement – was partially addressed by providing an executable implementation of norms: they are applied to capture the normative consequences of violations. While, as discussed by Vázquez-Salceda *et al.* [227], norm enforcement may be addressed from two perspectives – preventively by constraining unwanted behavior or reactively by detecting violations and responding to them – we strongly believe on the latter as the only viable approach to address e-contracting scenarios.

In this chapter we look into other means an EI may have at its disposal in order to enforce norm compliance. We will design and propose an abstract representation of the normative content of contracts in order to apply a domain-independent adaptive model of sanctions.

8.1 Rationale

A contract's normative structure will certainly reflect the coarse business workflow between the involved agents, but will probably include provisions for the most likely possible violations only. Further contingencies will often not be dealt with when establishing a contract, because it may be costly or even impossible to anticipate them. Even when relying on a preexistent normative framework (such as the one provided with our approach), contrary-to-duties may be poorly designed for some situations, or may simply be inexistent. This is when other coercive approaches may be relevant, in situations where agents try to take advantage of their potential gain when violating norms (because they might be more self-interested than socially concerned).

In the literature (e.g. [175][102]) we find, among others, two basic kinds of sanctions that an institution may apply in order to incentive norm compliance (or, to put it another way, to discourage deviations). *Direct material sanctions* have an immediate effect, and consist of affecting the resources an agent has (e.g. by applying fines). *Indirect social sanctions*, such as changing an agent's reputation, may have an effect that extends through time. Depending on the domain and on the set of agents that inhabit the institutional environment, the effectiveness of such sanctions may be different: if agents are not able to take advantage of other agents' reputation information, material sanctions should be used instead.

There are two general policies used when applying (direct) sanctions, which concern their intended effects: (i) *deterrence* aims at punishing the violator so as to discourage future violations; (ii) *retribution* aims at compensating the addressee of the violation. Bringing these policies to the electronic institution realm, we see retribution sanctions as those specified in contractual norms, be they negotiated or inherited from a preexistent normative framework. In this case the institution, while monitoring norm compliance, acts as a mediator. As for deterrence sanctions, they will be applied by the institution itself, and may be used so as to maintain order (by motivating agents to comply) and consequently trust in the system. A similar distinction is made by Fornara and Colombetti [78]: *active sanctions* describe actions to be performed by the violator (and if he does so the violation will become extinguished), while *passive sanctions* describe actions that the norm enforcer is authorized to perform.

Deterrence has also been studied from a different perspective in political science [241], where theories are proposed for explaining international relations in tense periods such as the Cold War. In this case, deterrence is based on threats that are made between different nations.

Economic approaches to law enforcement have suggested analyzing sanctions and their amplitude by taking into account their effects on parties' activities. Agents committing to norms that have associated deterrence sanctions enter risky activities, because they may unintentionally violate them. It has been argued [177] that under strict liability (where violators are always sanctioned) sanctions should equal harm done. An increase in the level of activity brings an increase in the expected harm; if damages equal harm, parties will have socially correct incentives to engage in risky activities (that is, to establish commitments). However, this conclusion relies on the additional assumption that parties are risk-neutral. If they are risk-averse, the optimal level of damages tends to be lower than harm. This comes from the fact that with risk-aversion, a sanction imposes a cost which does not exist under risk neutrality. As explained in [88], risk-aversion introduces costless deterrence and the policy-maker (an Electronic Institution in our case) should take that into account when choosing the optimal sanction.

The presence of social sanctions will also influence the behavior of agents concerning their commitments. Reputation-aware environments should have a lesser need for deterrence sanctions (see, for instance, [153]). Besides reducing agents' risk (see above), a reduction of deterrence mechanisms may be important for other reasons. On one hand, both the enforcement activities and the completion of direct sanctions may be costly, which asks for either lowering the resources used in those activities or eliminating sanctions in non-compensating cases. On the other hand, we can imagine (at least in theory) a computational system where these costs can be marginal: assuming that automatic norm monitoring is computationally inexpensive and that sanctions consist e.g. of fines that are debited from agents' accounts administered by the system. But in this case, higher fine levels require higher financial warranties from agents, which may once again decrease their level of activity: some agents may not meet such requirements, which will inhibit them from committing to certain normative relationships.

In this chapter we seek to explore these issues inside an institutional environment, under the following assumptions:

- Strict liability: norm violation is always detected.
- Costless enforcement: monitoring and sanctioning have a negligible cost to the institution.
- Unknown agent population: concerning agents' risk tolerance and social awareness.

We envisage a normative framework that is able to adapt itself (by changing applicable sanctions) according to some measurement of success, which will have to manage the following conflicting goals: i) keep the normative framework as simple as possible, by avoiding over-constraining the environment; and ii) maximize trust on the institutional environment's use. These conflicting goals must be balanced well enough in order to encourage agents to increase their level of activity, when the agent population's risk tolerance is unknown beforehand. Obviously, we assume that agents' preferences regarding the kinds of sanctions we employ are known: agents prefer not to be fined.

In order to develop an experimental model that allows us to test adaptation skills of an institutional normative environment, we will make use of an abstract model for contractual commitments. We then present a simple adaptation approach that consists on adjusting deterrence sanctions, based on which a set of experiments were drawn.

8.2 The Model

In our approach we take the stance that agents are truly autonomous, and thus cannot be forced to fulfill their obligations. The institution may, however, impose certain fines as deterrence sanctions: those fines are assumed to be fully regimented (that is, agents cannot escape them, e.g. because they were required, upon entering the institution, to make a deposit that is in control of the institution). Sanctions other than fines could also be envisaged as deterrence measures.

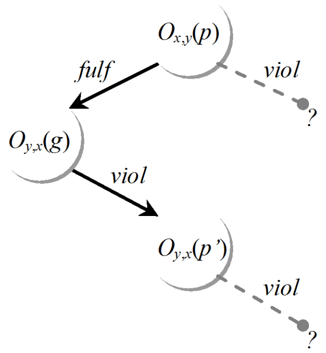


Figure 8.1: Sample commitment tree.

We are mainly concerned with e-contracting scenarios, wherein agents make mutual commitments and create business expectations. Violations, even when handled by contractual norms, should be seen as exceptional situations. Hence, if a certain kind of violation becomes frequent, response should be taken through an increase of sanctions.

8.2.1 Commitment trees

In order to obtain a tractable model for handling contractual commitments, we use a tree-based representation for interdependent obligations. This representation is useful for understanding the simulation model that we describe later on.

When establishing contracts, agents create a network of directed obligations, some of which are dependent on the fulfillment or violation of other obligations. For the sake of illustration, consider the following two-party contract: agent x will pay p currency units to agent y, after which y will deliver good g to x. In case y fails to deliver, he must return $p' = p + \delta$ to x. This sequence of commitments is illustrated in Figure 8.1, in a tree-like structure – a *commitment tree*. Each node (i.e., each commitment) represents a directed obligation from a bearer b to a counterparty c to bring about a fact f – $O_{b,c}(f)$ (unlike directed deadline obligations introduced in Section 6.2.1, for simplification we omit the temporal dimension). Bringing about the obliged fact is assumed to imply a cost to the obligation's bearer, and presumably produces some benefit to the counterparty (we will take this into account when providing a more formal account to the representation of nodes in the commitment tree). Each labeled directed edge in Figure 8.1 indicates, in the child node pointed to, what follows when the obligation contained in the parent node is fulfilled (*fulf*) or violated (*viol*). In this simple example nothing is specified should agent x violate his commitment to pay p, or should agent y violate his commitment to return p'. On the other hand, returning p' is seen as a sanction applied to y if he violates his obligation to deliver g.

Typically, a *viol* child node includes a contrary-to-duty that remedies the failure of the bearer to fulfill his previous obligation, potentially allowing for the contract to be resumed. A *fulf* child node will usually define a complementary obligation where the bearer and counterparty roles are switched.

While this example shows a simple binary tree, one can imagine multi-party contracts with a potentially complex commitment tree structure. The tree will not be binary if each obligation fulfillment or violation may lead to more than one consequence. Also, if we consider that a norm can prescribe an obligation if two or more fulfillments or violations occur, we end-up with a *directed acyclic graph* instead of a tree, since each node may have more than one parent. However, this is not very common in the case of violations (which are our main concern here): each violation will typically be handled in isolation (as in the model of "reparation chains" in [95]).

The violation of an obligation with a prescribed sanction may simply denote a case where an agent preferred to incur the sanction for matters of conflicting goals (e.g. he had another more important contract, and could not stand for both). If such violation becomes frequent, however, this may denote a flaw in the normative system that agents are being able to exploit to their own advantage.

8.2.2 Adaptation

The importance of adaptation in a normative framework resides in the fact that contracts may be unfair in certain execution outcomes. If self-interested agents exploit such flaws to their own profit, action should be taken in order to discourage such behaviors.

In order to build a model that adapts the normative framework in a domain-independent way, we will concentrate on adding deterrence fines to the system (which are not violable), instead of changing the prescribed obligations in each violation situation. The normative framework's adaptation is based on associating, with each obligation, a fine that can be strengthened or weakened (see Figure 8.2[1]). With this approach, every obligation will have a (possibly null) fine to be imposed on the bearer in case of violation; this fine is added up to the violation consequence in the child node already in the tree, if there is one.

In order to correctly model appropriate responses to specific situations, we need to assess how often an obligation is used, and how often it is violated. Fines will be updated according to these measurements. The basic principle that we rely on is that the strength of a fine should be directly proportional to its application frequency. As such, fines should increase when they are applied often, and decrease when they are not used. A low level of fine usage indicates that obligations are being fulfilled or they are not being used as often as desired: in both cases fines should be decreased, since they either are not needed or are

[1]From now on, we will only consider the case for *binary* commitment trees (this simplification does not limit the applicability of our approach, while it does make it easier to follow).

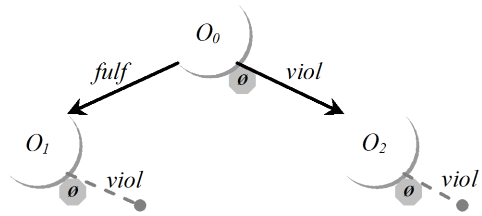


Figure 8.2: Binary commitment tree with null fines.

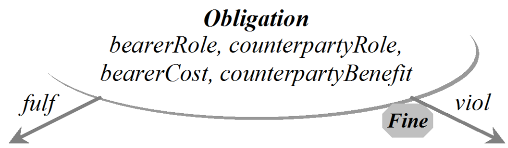


Figure 8.3: A node in a BCT.

inhibiting activity. On the other hand, a high level of fine usage means that agents still prefer to go through the sanction, and as such it should be increased as a deterrence mechanism. In summary, this approach tries to make fines (a) strong enough to discourage deviation and (b) weak enough to avoid unnecessary or counterproductive institutional control.

8.3 Simulation Environment

Aiming at the development of a simulation prototype that allows us to test the adaptation model briefly described above, we designed the following experimental scenario.

A number of agents will be in the environment, and each will be given the opportunity to sign a contract, whose structure is defined by the number of enacting roles and by an underlying binary commitment tree (BCT from now on). In the BCT structure, contract roles are used as bearers or counterparties of obligations. Furthermore, each obligation has an associated cost (to be supported by a fulfilling bearer) and benefit (to be collected by the counterparty of a fulfilled obligation). Therefore, when an obligation is fulfilled, the agent enacting the bearer role bears the cost of fulfillment, while the agent enacting the counterparty role gets the benefit. Figure 8.3 summarizes the characterization of a node in a BCT.

When an agent decides to sign a contract, he will enact the corresponding commitment tree with a role assigned to him before contracting. We say that the *state* of a contract enactment is the commitment currently under appreciation. If the bearer of such a commitment is the agent that decided to contract, he will be asked for a play: either to fulfill or to violate the commitment. If the commitment's bearer is not the agent, the system will decide whether the commitment will be fulfilled or not, according to a uniform strategy.

The current state will be updated according to the decision taken: if the choice is to fulfill, then the root commitment of the fulfillment sub-tree will become the current state; if the choice is to violate, then the root commitment of the violation sub-tree will become the current state. The contract terminates when the state becomes null (i.e. when no fulfillment/violation sub-tree exists upon a fulfill/violate decision).

8.3.1 Agent decision-making

Each agent has two distinct kinds of decisions to make. If he does not currently have an ongoing contract, he is given the opportunity to sign one. For that, a random role from the contract structure is selected and the agent is asked if he wants to contract with that role. Each agent is configured with a *risk tolerance* parameter $Rt \in [0; 1[$, which denotes his willingness to contract in the presence of violation fines. If $Rt = 0$, the agent will only decide to contract if he will be subject to no fines at all. On the other extreme, if $Rt \approx 1$, the agent will always risk to contract, regardless of any fines. An agent will decide to contract depending on the highest fine that is associated with commitments for the assigned role. In order to contract, the following relation should be true:

$$highestFine(role) \leq b * Rt/(1 - Rt) \tag{8.1}$$

where b is a slope parameter associated with the agent's budget.

We assume that agents always prefer to contract, regardless of commitment costs or benefits. A contract is presumably beneficial to all partners should they fulfill all their commitments. Having said this, we allow a contract to be unbalanced or incorrect from a safeness point of view, in the sense discussed in [58]. In our case, we consent that participating in the contract may in some cases be worse-off than not participating, depending on the behavior of contractual partners.

When an agent has an ongoing contract, whenever the contract's state is a commitment where he is the bearer he will decide whether to fulfill or to violate such a commitment. Depending on a so-called *in-contract strategy*, the agent will explore the contract's BCT in order to decide which option is best for him. Such strategies may vary from simply comparing the cost of fulfillment with the applicable fine in case of violation, to computing the path with the best outcome from the whole BCT. Some possible strategies will be presented in Section 8.4.1.

Agents are essentially expected utility maximizers. This means that, in principle, they will fulfill obligations only when the expected outcome from this choice is better than the expected outcome from violating (according to his in-contract strategy). We do however embed in our agents some notion of social welfare, which impels them to fulfill even when they do not have a strict advantage in doing so. While for now we do not consider the effect of reputation in future contracts, we allow in our model that agents are not all equally self-interested. For that we introduce a *social awareness* parameter $Sa \in [0; 1[$. If $Sa = 0$, the agent will violate whenever the outcome from this choice is better than the outcome from fulfilling. On the other extreme, if $Sa \approx 1$ the agent will always choose to fulfill. The agent will decide to fulfill an obligation O whenever the following relation is true:

$$violationOutcome(O) - fulfillmentOutcome(O) \leq b * Sa/(1 - Sa) \tag{8.2}$$

where b is as before. The violation/fulfillment outcomes are calculated by the in-contract strategy.

8.3.2 Fine update policy

In each simulation step, all agents running in the simulation will have a chance to play. After this, the contract structure will have a chance to adapt, that is, the fines associated to the BCT will be updated. Each fine is updated independently of all other fines.

In order to delineate a fine update policy, we first need to define the goal function that will be pursued. As mentioned before, fine updates should take into account how often they are applied. We define a threshold parameter $Th \in [0; 1]$ that roughly indicates the highest percentage of fines that the system should accept as normal. For instance, with a value $Th = 0.1$ we are saying that if more than 10% of the agents running in the simulation violate a given obligation the normative system will raise the fine in the next step – in this case, we say that 10% of the total number of agents is the number of *tolerated* violations. Furthermore, since not all agents will be in the same state at a given time point, we adjust the threshold according to the number of agents that did in fact make a decision concerning the fulfillment or violation of a specific obligation (because they were in that state). For instance, if with a group of 1000 agents we have 10 violations of a specific obligation in a simulation step, this may have a different response from the normative environment depending on the number of agents that went through that same obligation at that time step. If there were 10 play decisions taken on that obligation, this makes a 100% percentage of violations; if there were 100 plays, that percentage comes down to 10%. While in none of these cases we exceed 10% of the total number of agents (1000), it seems clear that the system should react in the former case.

The fine associated with each state will be increased if the number of violations exceeds the following tolerated violations function:

$$toleratedViolations = 2 * Th * Nag/(1 + e^{-(5/Nag)*x}) - Th * Nag \tag{8.3}$$

where Nag is the number of agents running in the simulation and x is the number of agents that were in this state. This is a sigmoid function with an upper bound set at $Th * Nag$ (a percentage of the total number of agents). The steepness parameter is $5/Nag$, which makes the sigmoid curve approach the upper bound close to Nag, which is the ceiling for x (there can be no more than Nag agents at this state).

The fine will be decreased whenever the number of violations does not exceed the number of tolerated violations. Fines are increased heavier than they are decreased. We have set an increase step of 0.1 and a decrease step of 0.01. This fixed update policy determines the convergence rate for fines. Furthermore, fines will be applied rounded to the first decimal place, which gives a sense that it takes ten simulation steps (without exceeding the tolerated violations function) to decrease the fine value.

8.4 Experimental Evaluation

In this section we provide, through a set of experiments, an evaluation of the adaptation model using the simulation environment described above.

8.4.1 Settings

What we want to study with the simulation scena-rio described in the preceding section is whether the normative framework is able to adapt and stabilize fine changes in a situation with a static agent population. Furthermore, the system should keep fines as low as possible, while still conforming to the goal function outlined above. This is because the system aims to avoid excessive control and through that maximize agents' contracting activity, which should be obtained with less risk exposure in an agent population with unknown risk tolerance.

If we change the agent population in the middle of the simulation, then we have a moving target setting, which is out of the scope of the experiments we report in this chapter. However, since we lower fines whenever the tolerated violations are not exceeded, we believe that the system will quickly adapt in a moving target setting.

Contract structures

Since we are not concerned with the correctness of the contract to be signed, we may abstract away from the concrete meaning of the contract that is represented by a BCT. In other words, we may carry out experiments with a large number of arbitrary BCTs. Figure 8.4[2] shows some possibilities, all considering two roles

[2]For simplicity, fines are not shown; however, every node (including leaf nodes) should be seen as shown in Figure 8.3.

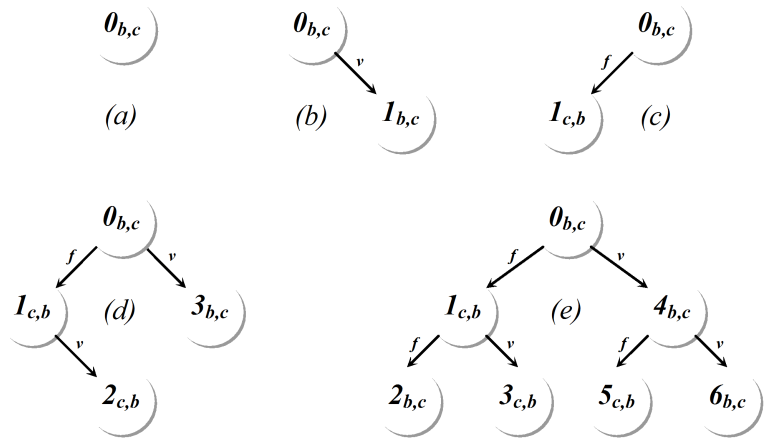


Figure 8.4: Binary commitment trees: each node $Id_{i,j}$ is an obligation, where i is the bearer and j is the counterparty.

only. For instance, (d) includes two complementary obligations 0 and 1, and their respective *contrary-to-duties* 3 and 2. We shall call obligation 1 the *to-duty* obligation of obligation 0.

We will present some experimental results based on some of these BCTs. In all cases, obligation costs were set at 10.0 and benefits at 12.0 (setting benefits higher than costs tries to give all partners some gain when the contract is well-balanced and is smoothly enacted). Also, fines were initialized at 0.0.

Agents

As noted before, we aim at testing the normative framework's adaptation when the agent population is unknown, concerning agents' risk tolerance and social awareness. For that reason, all agents in the system have a uniform random distribution con-cerning the risk-tolerance and social-awareness parameters. Also, for these parameters the slope value b was set to 10.0. This makes the right hand side of Equation 8.1 reach 10.0 when a middle value of 0.5 is used for Rt. It also turns out to make 10.0 a ceiling for fines.

Several in-contract strategies can be devised, representing different reasoning abilities of agents when deciding whether to fulfill or violate an obligation. As explained before, the in-contract strategy will be used to compute the fulfillment and violation outcomes at a given state. We consider the following simple strategies, which may have different relevance depending on the BCT being used:

i ***Local***: considers only local information with respect to the obligation being analyzed.
<u>Fulfillment outcome (FO):</u> – fulfillment cost
<u>Violation outcome (VO):</u> – fine

ii ***LocalCtd***: considers the cost of fulfilling the contrary-to-duty obligation (if there is one); ignores possible entitlements in case of fulfillment
FO: – fulfillment cost
VO: – fine – contrary-to-duty cost

iii ***LocalTd***: considers the benefit to gain from the to-duty obligation's fulfillment (if there is one); ignores possible normative sanctions in case of violation
FO: – fulfillment cost + to-duty benefit
VO: – fine

iv ***LocalBoth***: a mixture of *LocalCtd* and *LocalTd*
FO: – fulfillment cost + to-duty benefit
VO: – fine – contrary-to-duty cost

v ***FulfillmentBalance***: considers the balance (net gain) obtained if the contract is enacted without any violations
FO: net gain if every participant fulfills the contract
VO: – fine

vi ***DoubleFulfillmentBalance***: considers two possible balances, one as in *FulfillmentBalance* and another by assuming that there will be no further violations from the contrary-to-duty obligation onwards (in this case the agent is fined)
FO: net gain if every participant fulfills the contract
VO: net gain if every participant fulfills the contract from the contrary-to-duty obligation onwards – fine

vii ***BestPathCompliantPartners***: explores the whole BCT in order to find the best net gain for every possible path, assuming that *contract partners will always fulfill*
FO: best net gain from the fulfillment subtree – fulfillment cost
VO: best net gain from the violation subtree – fine

viii ***BestPathMinimax***: explores the whole BCT in order to find the best net gain for every possible path, considering that *contract partners will use the same strategy*
FO: best net gain from the fulfillment subtree – fulfillment cost
VO: best net gain from the violation subtree – fine

Strategies iii through vii assume that partners will always fulfill their obligations. Analyzing these strategies together with the BCTs depicted in Figure 8.4, we can see that, for instance, *FulfillmentBalance* will only make sense in tree (e), since in all other BCTs the same outcome can be achieved with less computationally demanding strategies.

Strategy viii is a *minimax* strategy: the agent will maximize his own expected utility while assuming that the other agent will do the same. For instance, considering BCT (d) at Figure 8.4 with no fines, the agent will choose to violate on every obligation. While this seems obvious for obligations 1, 2 and 3 (there is no personal benefit in fulfilling), in obligation 0 the agent chooses to violate because he assumes that the counterparty will violate on 1 and 2, bringing him no benefit that can compensate the cost of fulfilling on 0. This strategy seems counterintuitive with the very decision of establishing a contract. However, for the sake of testing the adaptation capabilities of the normative framework, this agent decision practice is bearable.

8.4.2 Experiments and results

In all experiments a uniform strategy "always fulfill" was used by the system for commitments whose bearer is not a simulation agent. The violation threshold parameter *Th* was set to 0.1. Each simulation was run with 10000 agents and 1000 time steps.

Figures 8.5-8.14 present the evolution of fines and their effect on agents' behavior for some possible combinations of BCTs and in-contract strategies.

In BCT (c) a *LocalTD* strategy (Figure 8.5) is able to grab the benefit achieved from obligation 1 when fulfilling obligation 0. Only the violation of obligation 1 is tempting, and thus the system adapted the corresponding fine. The concrete value obtained for this fine is correlated with the values defined for the obligation cost and benefit, together with the strategy used by agents when deciding on fulfilling the obligation. Figure 8.6 shows the relative cumulative aver-age of violations with these settings. We can observe a decrease on the number of violations for obligation 1 as a consequence of the fine increase, which ceases when the number of violations is below the tolerated violations threshold (see Equation 8.3).

In BCT (d) the *LocalTD* strategy (Figure 8.7) impels agents to fulfill obligation 0. Agents that are more socially concerned will tend to fulfill obligation 1 with lower fines than other agents, hence the difference between fines 1 and 2. Figure 8.8 shows the corresponding relative cumulative average of violations. The effect of fines on agents' behavior is clearly visible.

In the same scenario, *BestPathMinimax* (Figure 8.9) gives agents the ability of evaluating every possible outcome with rational plays for both contractual partners – an agent will maximize his own expected utility while assuming that the other agent will do the same. For BCT (d) this means that each agent playing in state 0 initially violates because he sees his partner preferring to violate obligation 1 (and 2), therefore giving him no benefit. However, when fines 2 and 3 are high enough, fines 0 and 1 are no longer necessary. Figure 8.10 shows the evolution of violations for this case. There are no violations for obligations 0 and 1 (their averages tend to 0 in the figure) since before time step 400 (fines applied in these states become nil).

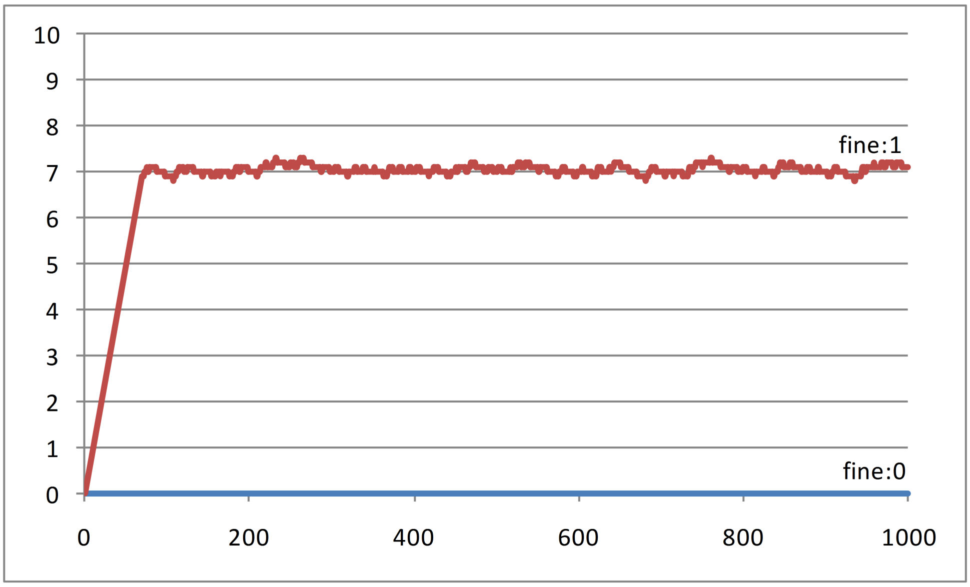


Figure 8.5: Fine evolution for BCT (c) and *LocalTD*.

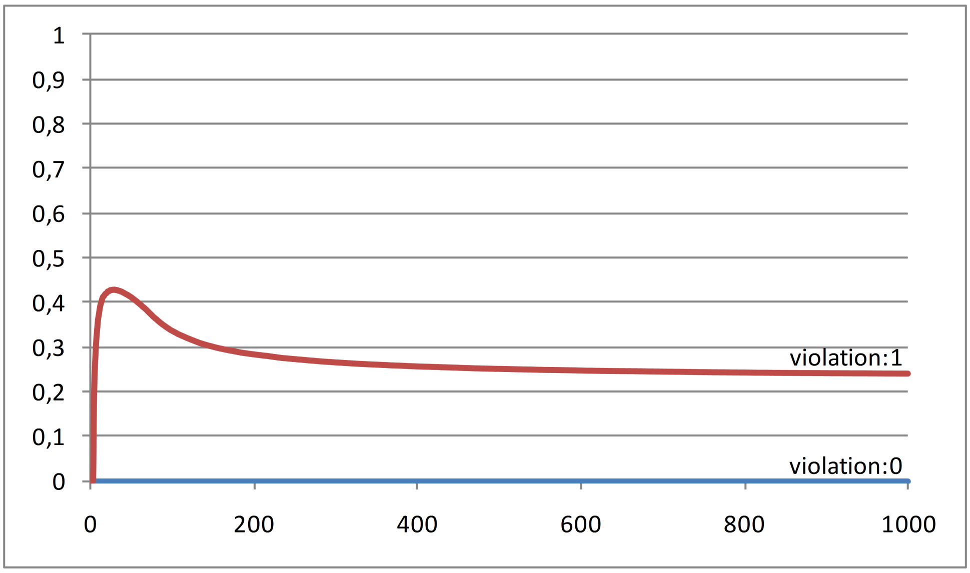


Figure 8.6: Violation cumulative average (%) for BCT (c) and *LocalTD*.

As for BCT (e), unlike the previous two contractual structures, this one is not profitable (with the complete fulfillment execution 0-1-2) for the agent playing at state 0, if we consider the values set for every obligation's cost (10.0) and benefit (12.0). The *DoubleFulfillmentBalance* strategy (Figure 8.11) is able to detect the better path 0-4-5, causing a reaction of the normative system with a raise of fine 0. Without this escape, violating obligation 2 is a means of taking some profit out of the game, bringing a raise of fine 2. Figure 8.12 shows the evolution of violations

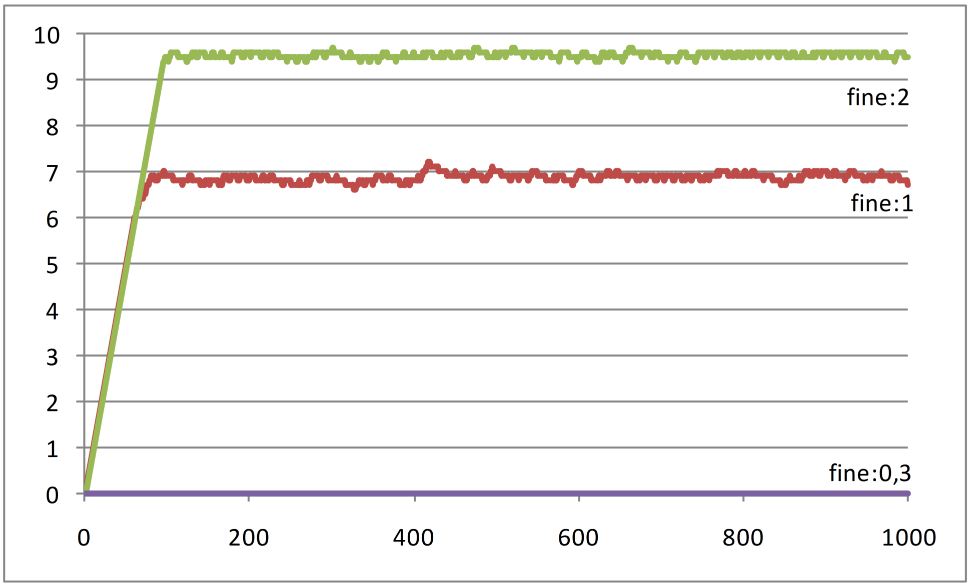


Figure 8.7: Fine evolution for BCT (d) and *LocalTD*.

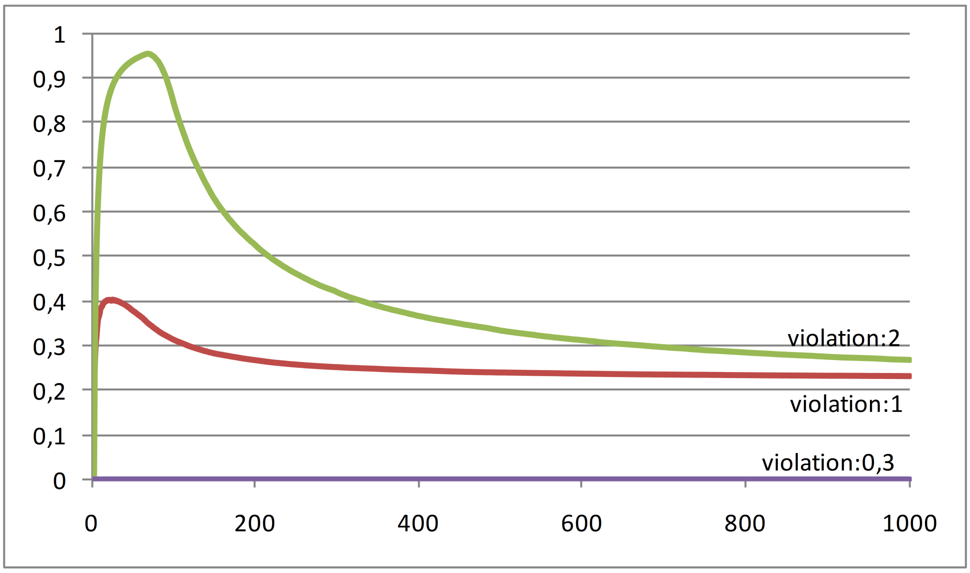


Figure 8.8: Violation cumulative average (%) for BCT (d) and *LocalTD*.

for this case. The delayed raise on the number of violations for obligation 2 is clearly visible.

The *BestPathMinimax* strategy in this scenario (Figure 8.13) makes up the most complex setting we have experimented. Starting at state 0, the best path when playing with a similar agent is initially to violate all obligations (including contrary-to-duties), bringing an outcome of 0. This is because the agent assumes that his partner will maximize his own profit, therefore preferring to violate at

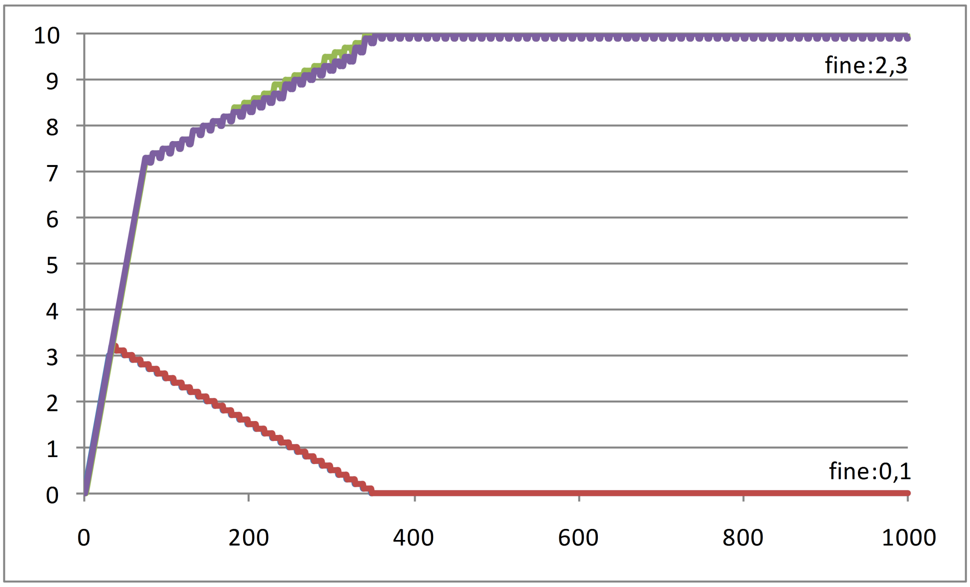


Figure 8.9: Fine evolution for BCT (d) and *BestPathMinimax*.

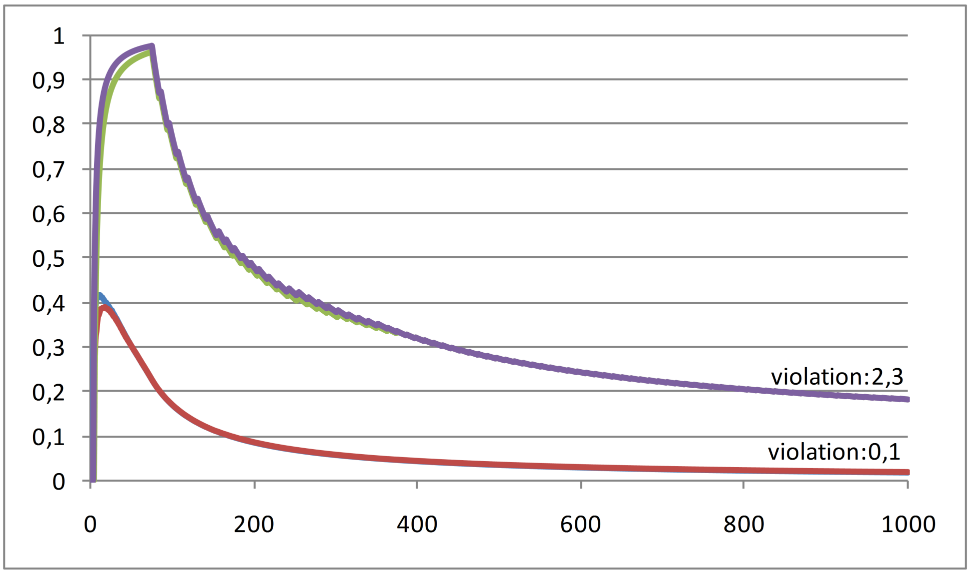


Figure 8.10: Violation cum. average (%) for BCT (d) and *BestPathMinimax*.

states 1 and 5. Fines at obligations 0 and 1 become ineffective as soon as fines associated with contrary-to-duties are high enough. Figure 8.14 shows the corresponding evolution of violations.

The system is able to adjust deterrence sanction values to the behavior of an agent population with any combination of BCTs and in-contract strategies, stabilizing fines after a period of time. We should emphasize that the system continuously tries to lower fines, which is observable by the slight fluc-tuations of

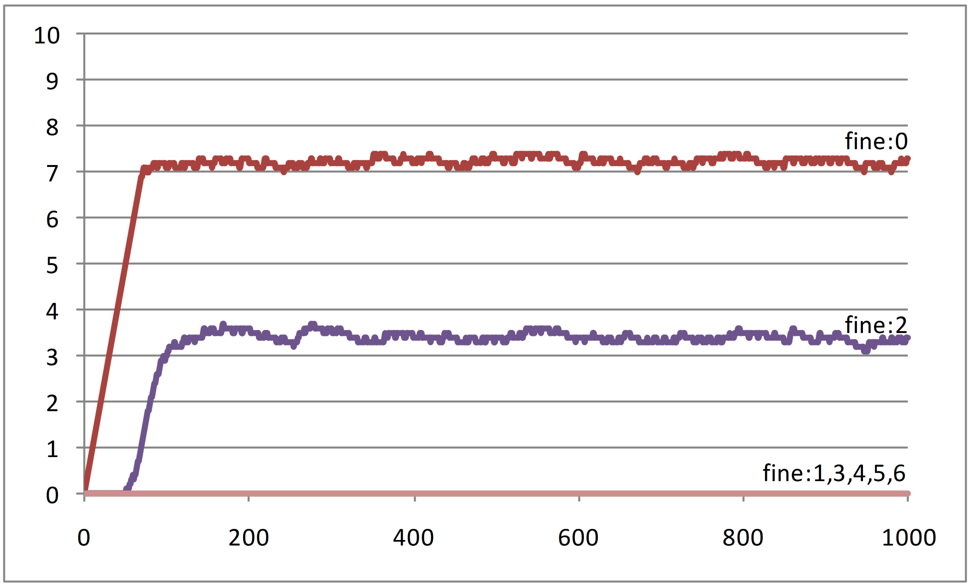


Figure 8.11: Fine evolution for BCT (e) and *DoubleFulfillmentBalance*.

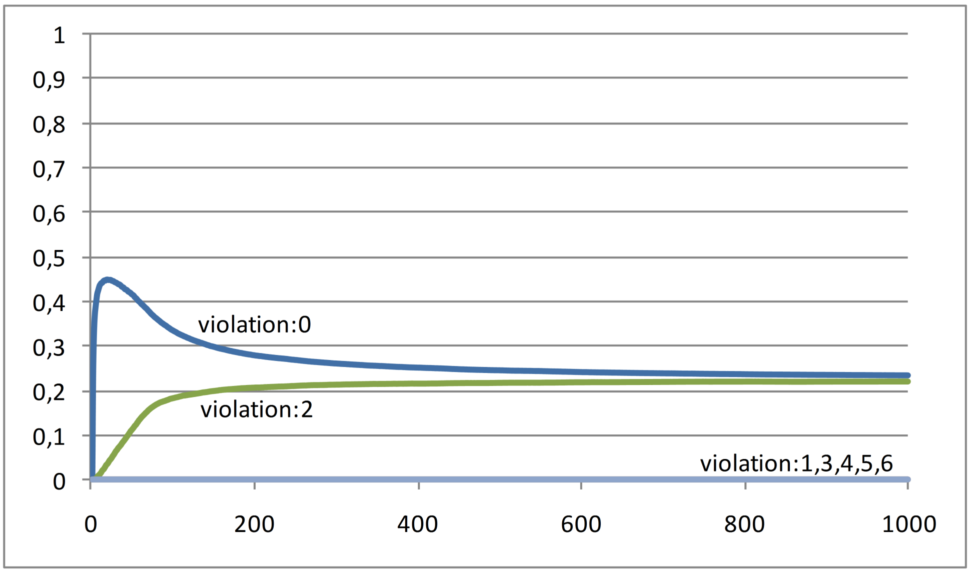


Figure 8.12: Viol. cum. average (%) for BCT (e) and *DoubleFulfillmentBalance*.

fines towards the end of curves in figures 8.5, 8.7, 8.9, 8.11 and 8.13. Therefore, system imposed fine levels are the lowest that keep violations below the tolerated violations function.

Besides affecting the number of violations (that is, agents' in-contract behavior), the adaptation of fines also affects contractual activity. Figure 8.15 shows the relative cumulative average of contracts in each of the described settings. While analyzing this graph, it is clear that when agents use the *BestPathMin-*

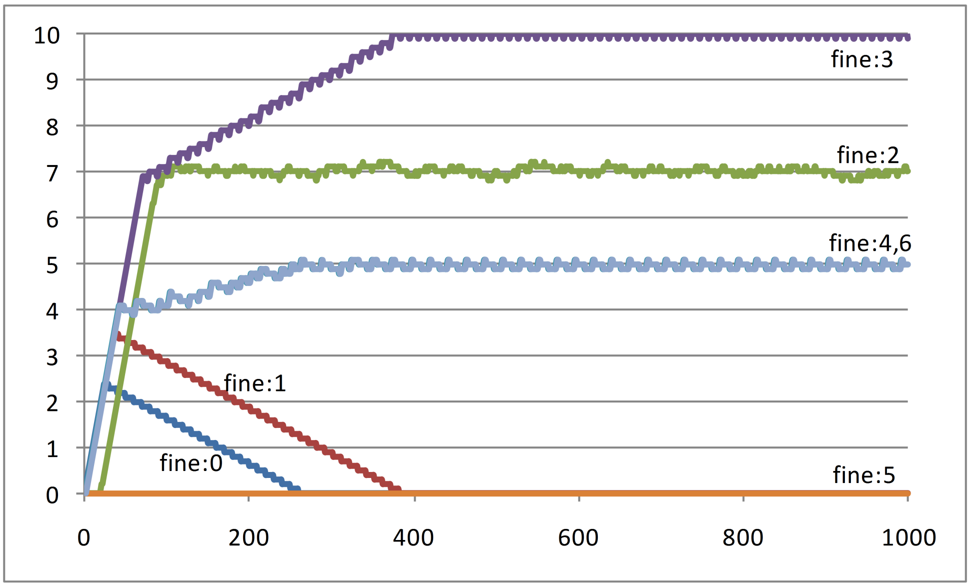


Figure 8.13: Fine evolution for BCT (e) and *BestPathMinimax*.

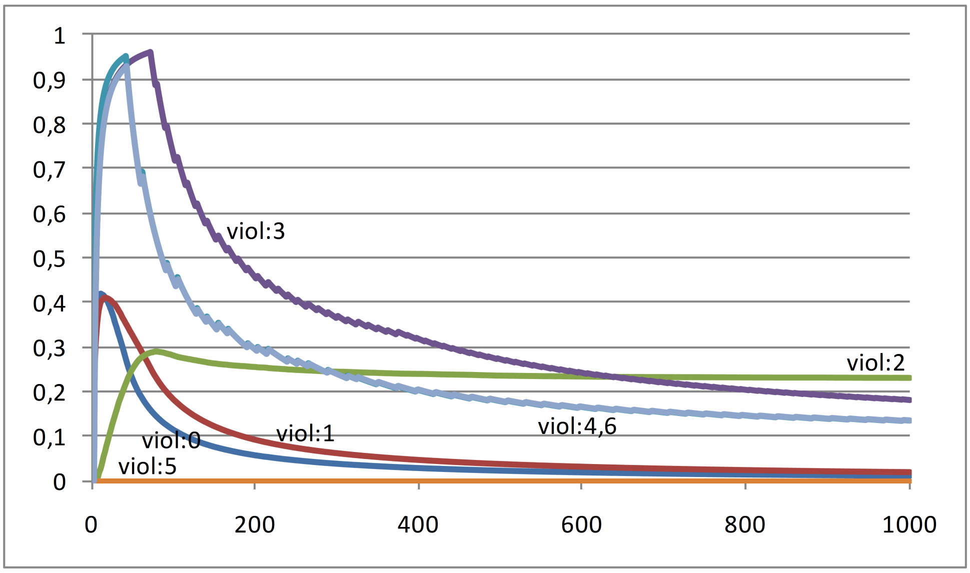


Figure 8.14: Violation cum. average (%) for BCT (e) and *BestPathMinimax*.

imax strategy they pose further demands on the system when trying to force agents to comply. In other words, agents use more information and therefore the adaptive mechanism must increase the level of sanctions in order to prevent excessive violation levels. As a consequence, agents that are more risk-averse tend to lower their level of activity, which explains why the number of contracts in these cases is lower.

We should also note that in these scenarios, where agent populations have no

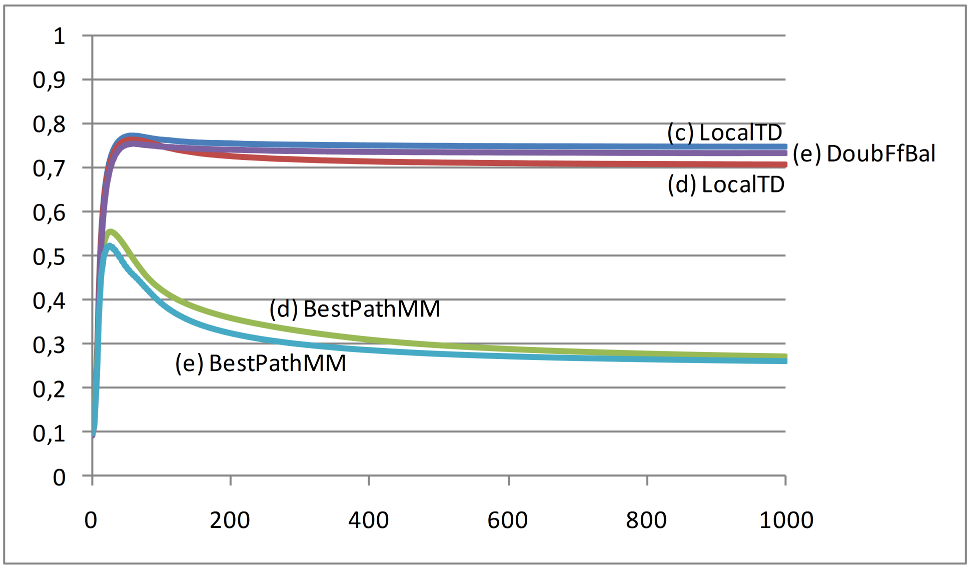


Figure 8.15: Contract cumulative average (%) for the five settings.

bias regarding social awareness or risk tolerance, the potentially harmful effect of fine adaptation in contractual activity is marginal for the remaining in-contract strategies.

8.5 Addressing Different Agent Populations

In this section we analyze the adaptation of the system when handling different agent populations, in which risk tolerance and social awareness distributions are concerned. For that, a set of experiments were conducted using BCT (d) (see Figure 8.4). The *BestPathMinimax* strategy was used by all simulation agents. The reader may want to observe Figure 8.9 again in order to recall the system's behavior when addressing a uniform distribution of agents (concerning risk-tolerance and social-awareness).

8.5.1 Risk tolerance

With this first group of experiments we aimed at observing the behavior of the deterrence sanction adaptation model when facing agent populations with different risk tolerance distributions. In a population that tends to be more risk-averse, higher fines should tend to decrease. In these experiments we used beta distributions centered at different risk tolerance values, in order to represent populations having a predominance of agents with specific risk tolerances. For each beta distribution, we set $\alpha = 1 + (c * p - c)$ and $\beta = p - (c * p - c)$, where c is the center value and p is a peak factor that we have set to 100.

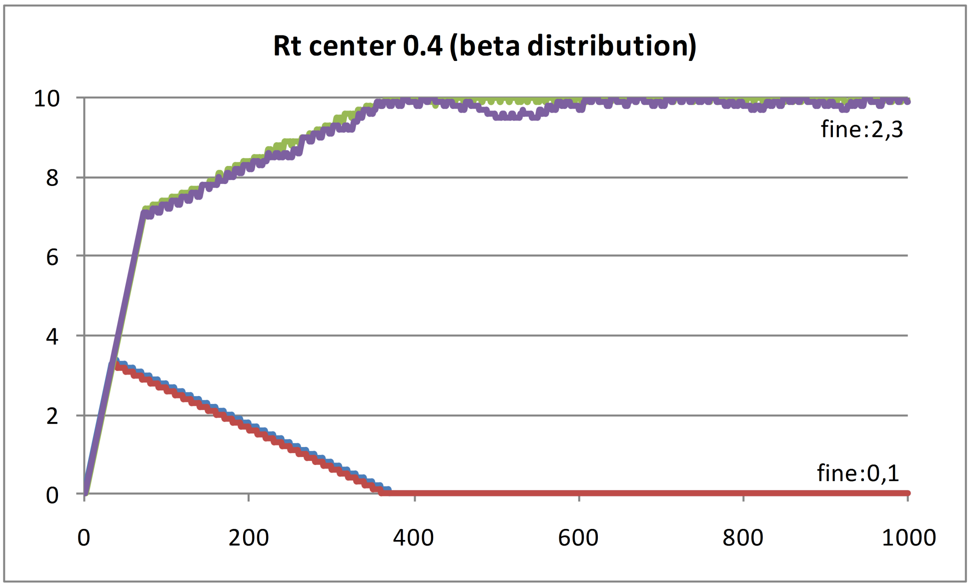


Figure 8.16: Fine evolution for BCT (d) and *BestPathMinimax*: beta distribution of risk tolerance centered at 0.4 and uniform distribution of social awareness.

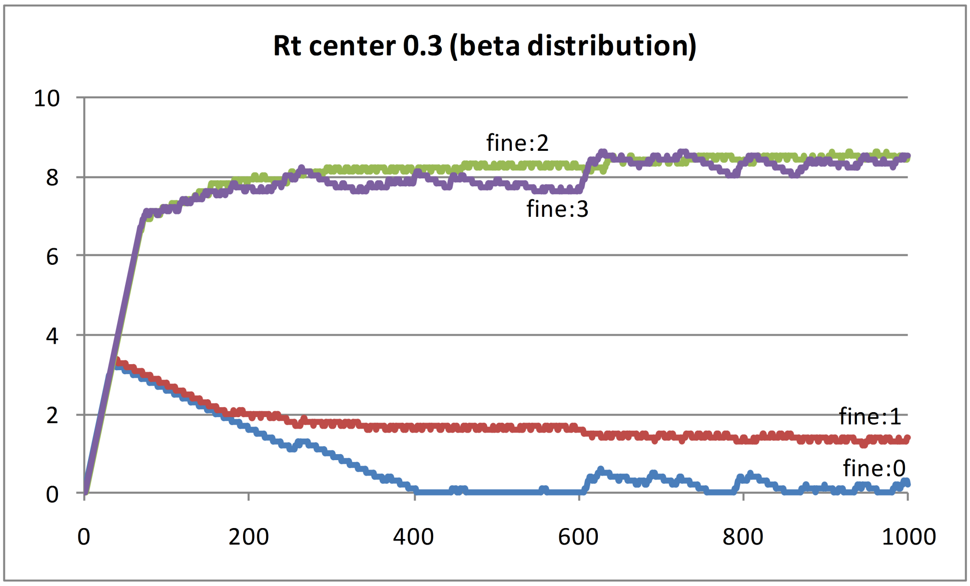


Figure 8.17: Fine evolution for BCT (d) and *BestPathMinimax*: beta distribution of risk tolerance centered at 0.3 and uniform distribution of social awareness.

Figures 8.16-8.19 show fine evolutions for different risk tolerance center values. As expected, higher fines tend to decrease with lower risk tolerance values. This is due to the fact that, when deciding whether to contract or not, agents compare their risk tolerance with the highest applicable fine.

Another interesting observation is that while the highest fines tend to decrease,

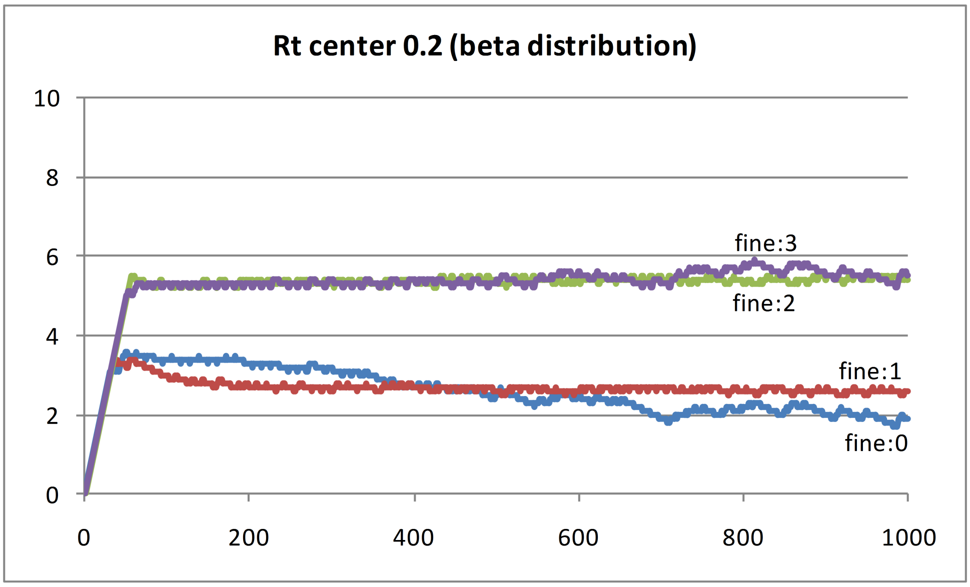


Figure 8.18: Fine evolution for BCT (d) and *BestPathMinimax*: beta distribution of risk tolerance centered at 0.2 and uniform distribution of social awareness.

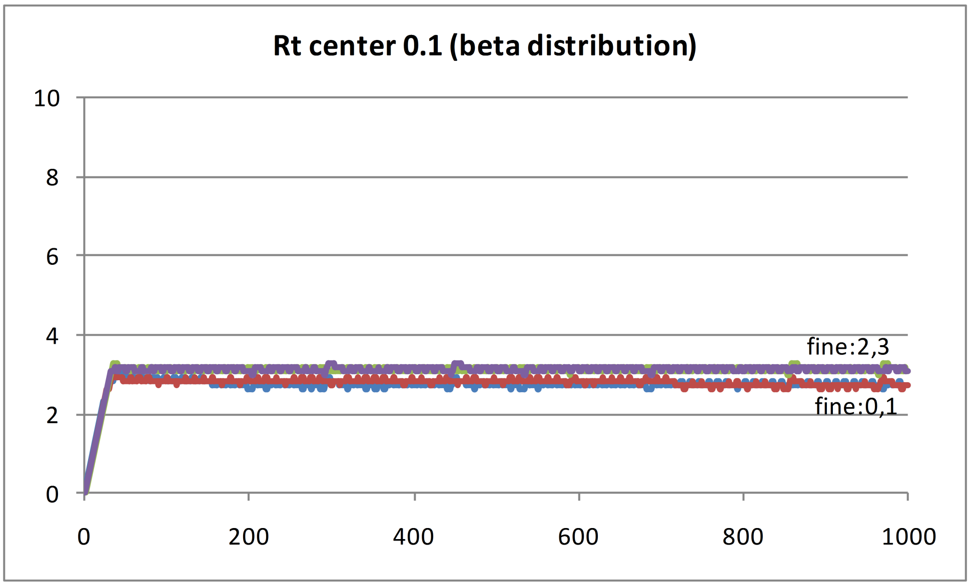


Figure 8.19: Fine evolution for BCT (d) and *BestPathMinimax*: beta distribution of risk tolerance centered at 0.1 and uniform distribution of social awareness.

the system tries to compensate this potentially lower ability to ascertain the desired level of compliance by increasing other sanctions. More specifically, since fines 3 and 2 are lowered, they lose their effect on decisions taken at states 0 and 1, respectively. As a consequence, fines in these states are raised.

This outcome turns out to be an important emergent property of the normative system: the ability to grasp interdependencies between fines applied to different nodes in the BCT, without being preprogrammed to do so (the fine update policy adapts fines in an independent way). Furthermore, such interdependencies are caused by the in-contract strategy used by agents; if agents do not take into account possible "future" fines when making a decision (as with strategies i to vi introduced in Section 8.4.1), then the system behavior will not pointlessly make a connection between fines.

8.5.2 Social awareness

With this second group of experiments we aimed at observing the behavior of the deterrence sanction adaptation model when facing agent populations with different social awareness distributions. In a population that tends to be more socially concerned, fines should tend to decrease. Selfish agents will only fulfill if it is in their own interest, while a higher social awareness impels agents to fulfill even when they do not benefit directly from that option.

Figures 8.20-8.21 show fine evolutions for different social awareness center values (using beta distributions as before). As expected, fines tend to increase with lower social awareness values. By doing so, the system tries to discourage commitment violations. The dependency mentioned before between fines is also visible here: fines 3 and 2 tend to absorb the effects of fines 0 and 1 sooner for higher social awareness values, and the system is able to find these intricacies.

8.6 Combining risk tolerance and social awareness

By adjusting both parameters when setting up an agent population, we get a combination of the effects identified above. Figure 8.22 shows what happens when we set both risk tolerance and social awareness to beta distributions centered at 0.1. In this case, since highest fines are limited by a low risk tolerance, the system raises fines 0 and 1 as much as it can, in order to try to force a population of mostly self-interested and risk-averse agents to contract and also comply with contractual commitments.

We should add that in these extreme and unlikely conditions the normative system is not successful: the obtained fine levels are insufficient to force compliance, and at the same time too demanding to motivate contractual activity. This means that the few agents that do contract (which nevertheless are in essence risk-averse) will violate their commitments (because they are also too self-interested).

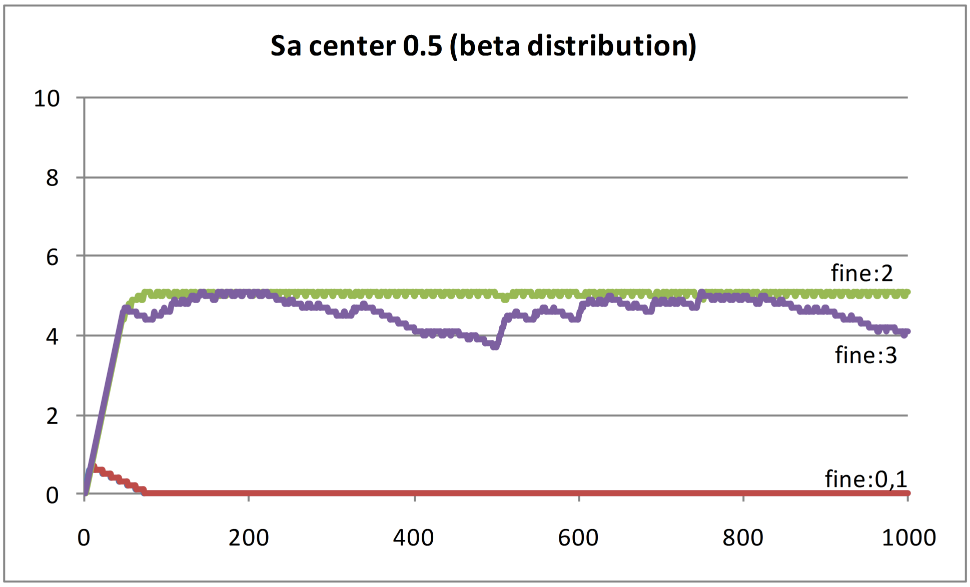


Figure 8.20: Fine evolution for BCT (d) and *BestPathMinimax*: uniform distribution of risk tolerance and beta distrib. of social awareness centered at 0.5.

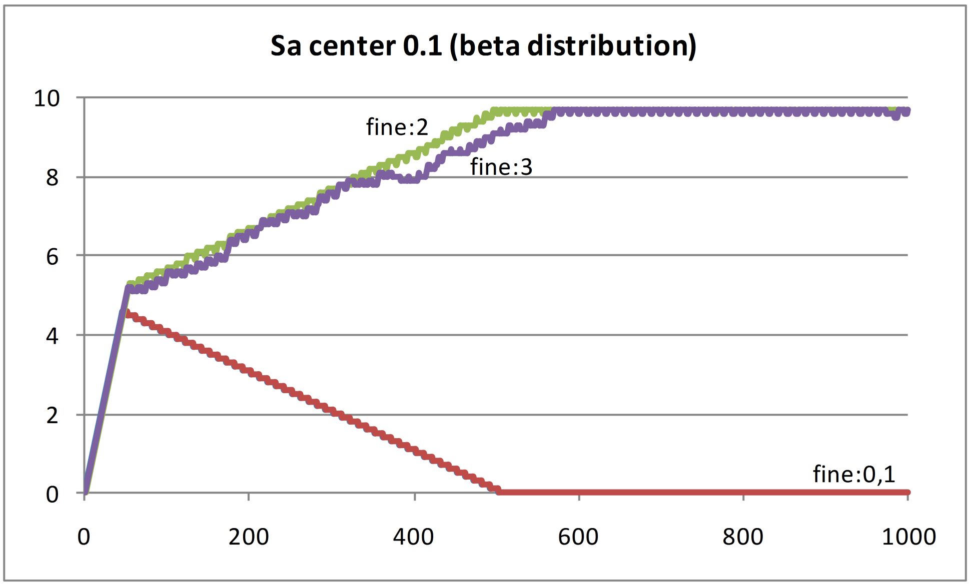


Figure 8.21: Fine evolution for BCT (d) and *BestPathMinimax*: uniform distribution of risk tolerance and beta distrib. of social awareness centered at 0.1.

8.7 Discussion

Embedding adaptive enforcement mechanisms in normative frameworks is important in open environments. Adapting deterrence levels to the behavior of an agent population is important when the normative space has imperfections that make

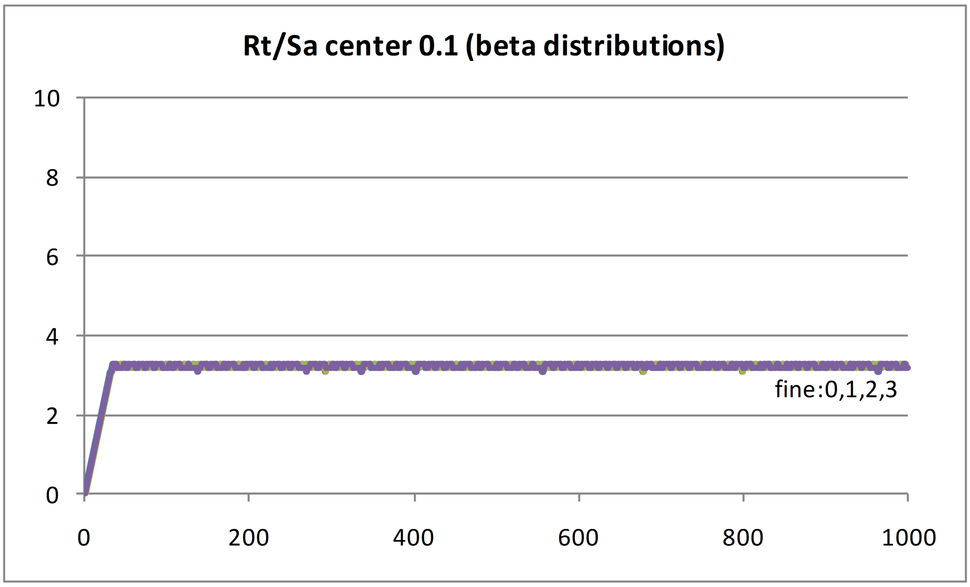


Figure 8.22: Fine evolution for BCT (d) and *BestPathMinimax*: beta distributions of risk tolerance and social awareness centered at 0.1.

contracts to which norms apply unfair, opening the possibility for self-interested agents to exploit their potential advantage.

In this chapter we have presented a simple model for the adaptation of deterrence sanctions used in a normative framework. We have shown that it is feasible to adapt deterrence levels to the behavior of an agent population: under uniform random distributions the system is able to adapt by appropriately raising and stabilizing fine values. This adaptive capability can be included in the set of the major contributions of this thesis. We have built an abstraction for contractual commitments by modeling their corresponding obligations in a binary commitment tree structure. In such a tree we are able to include both "to-duty" complementary obligations and contrary-to-duty retribution sanctions. This abstract representation allows us to consider contracts of arbitrary complexity.

This preliminary research on enforcement mechanisms to be included in an institutional normative environment has been published in [142] and [145].

8.7.1 Characterizing agent behavior

We have studied how an adaptive deterrence sanctioning model, while trying to "maintain order", responds when facing different agent populations. Such populations were characterized by a predominant level of risk tolerance and social awareness. The use of a social awareness parameter tries to take into account heterogeneous social attitudes. Configuring agents with different risk attitudes was inspired by the economic theory on deterrence sanctions [177], stating that agents incur a risk when making contracts that are subject to deterrence sanctions.

The parameterization of agents with different social attitudes is common in computational models for social interactions. Boella and van der Torre [22] distinguish three types of agents, according to how they are able to deal with norms. *Norm internalizing* agents shape their decision-making through the adoption of norms: they no longer waste any time deciding on whether to fulfill a norm once it has been internalized. *Respectful* agents fulfill obligations simply because obligations should be fulfilled, irrelevant of there being associated sanctions in case of violation (Castelfranchi [34] considers that the presence of external control through sanctions is a sub-ideal situation). Finally, *selfish* agents are those that fulfill obligations only if they prefer to do so, i.e., if the situation resulting from fulfilling is preferred (taking into account applicable sanctions in case of violation) according to their own goals and desires.

Norm internalization has been studied in social sciences, and also in the context of the EMIL project [5]. Norm internalization is seen as a less costly and more reliable enforcement mechanism than social control. By internalizing a norm, agents detach norm fulfillment from any outside enforcement mechanisms.

Andrighetto *et al.* [4] present an architecture for normative agents that includes normative-BDI notions. A *normative belief* (created by a norm recognition process) is a mental representation that a given action is either obligatory, forbidden or permitted. A *normative goal* (created by a norm adoption process) is an internal goal relativized to a normative belief: the goal will be pursued because the underlying action is believed to be prescribed by a norm. Conte *et al.* [41] further extend this work to identify the mental underpinnings of norm internalization processes. They claim that compliance is more robust if norms are internalized: compliance becomes independent of external sanctioning entities. Furthermore, norm internalizing agents are supposed to be much better at defending the norms because they have an incentive to punish agents who do not comply with them. This, in turn, is seen as an indispensable tool for distributed social control. However, some researches (e.g. [201]) advocate that these virtues are only long lasting if an external sanctioning system is never completely abandoned, basing their arguments on reinforcement learning theory.

8.7.2 Social control

Our experimental evaluations show that imposed fines tend to be lower when agents are more risk-averse or more socially concerned. We also observed that when a combination of sanctions is able to drive agents to comply with their commitments, the adaptive mechanism is able to pursue such a combination when constraints limit some options – such constraints are rooted in the agent population (namely in the predominant risk tolerance), and are implicitly captured in the fine update policy. This ability is an interesting emergent property of the system.

Influencing agent decision making regarding social commitments is generally conceived as *social control* [34], and is usually focused on enforcement, sanctions and reputation. A different perspective has been taken by Brafman and Tennenholtz in [25], where some agents in the system are directly controlled by the system's designer. Making such agents play specific strategies will lower the payoff of joint activities when uncontrolled agents play selfishly, therefore making them choose to fulfill. This seems unrealistic in contracting scenarios. Yet, the authors have made a theoretical analysis in scenarios where uncontrolled agents are expected utility maximizers and when they are reinforcement learners. Such scenarios can be tested in our simulation model as well.

Dynamic properties of normative systems have been studied from different perspectives. In [204], Sen and Airiau look at norms as patterns of behavior that may emerge bottom-up from agent interactions. In our case, however, the normative system is external to the agents, and we seek to adapt it to a specific agent population in order to pursue an overall system goal.

Sanction-based self-adaptation of institutional normative environments is also studied by Campos *et al.* in [31], with two significant differences to our approach. First, their adaptation model is based on the definition of domain-dependent transition functions, stating what specific change should be made in a specific norm when some goal specification is not met. Second, their model does not assume strict liability: agents are able to violate norms while not being detected.

In the work presented in this chapter we have not considered the influence of reputation on agent's contractual behavior. It has been argued [153] that in the presence of reputation mechanisms there is a lesser need for deterrence policies. We believe that positive reputation updates triggered by the normative environment may be an incentive for agents to fulfill their commitments. The interplay between a normative environment and computational trust and reputation measures lies in the core of the Electronic Institution Platform project, as described in Chapter 4.

Chapter 9

Conclusions

The research work carried out in this thesis was centered around the role electronic institutions can play in agent-based e-contracting. As the use of electronic means for conducting business relations matures, the need for automated tools that address e-contract handling will likely increase. At present times, the development of such tools presents many technological, business and legal challenges [6]. Nevertheless, cutting-edge research efforts are being made to address a vision of the future Internet, where the number of business opportunities that are available will increase exponentially. This will inevitably ask for automated tools, which may well result from research efforts within the MAS research community.

This chapter looks back to the main research contributions of this thesis, trying to comment on their virtues and shortcomings. We will also comment on the potential application of our contributions in other domains, and outline some lines of future work.

9.1 Main Achievements

In this section we will recover the research questions outlined in Section 1.3.1 and explain how the research carried out has made its way in providing answers to these questions.

The concept of electronic institution has been studied by several authors. However, in our view none of the previous approaches has been directed towards the application domain of e-contracting. This is why we have formulated the first research question, which we here reproduce:

- *How can the concept of Electronic Institution be adapted to address the domain of agent-based e-contracting?*

In respect to previous approaches, we have provided a new perspective on the Electronic Institution (EI) concept, by giving it a more comprehensive nature.

The EI is no longer a fully controlled environment implementing a predefined interaction scheme (as in [68]), but instead includes a normative environment whose normative framework, up to a certain extent, evolves through time as agents establish contracts. Furthermore, the EI is seen as a software platform providing a set of services that are articulated to provide assistance on agent-mediated e-contracting activities, such as negotiation, contract drafting, contract monitoring and enforcement.

Looking at the main subject of this work, which concerns the handling of e-contracts (from their creation to their monitoring and enforcement), we get to the core component of the EI, which consists of its normative environment. This leads us to the second research question:

- *What kind of e-contracting support should we embed in a normative environment?*

Two issues were addressed under this question: e-contract establishment and monitoring. We observed that given the current state-of-the-art it is not likely to have autonomous software agents creating e-contracts fully from scratch.

We have looked at what legal theory tells us about the use of contract clauses applied in business, and have followed two main notions. The first is *contractual freedom*: in principle, business partners are free to create contracts of arbitrary content. The second is the notion of *default rules* as used in contract law: predefined norms should be seen as facilitating rather than constraining contractual activity.

Based on these assumptions, we have designed a model for norm inheritance and defeasibility that enables the specification of a normative framework that can address a rich set of contracting situations. We have illustrated this specification both with one-off business cases (which were represented by contracts of sale) and standing agreements (where parties establish cooperation agreements that span several transactions over time). The normative framework is meant to facilitate e-contract drafting, by providing a background normative core for specific types of e-contracts. This alleviates the technological challenges raised by automated e-contracting.

When considering the employment of this infrastructure in the real-world, a major effort should be concentrated on knowledge engineering of norms applicable to different business contexts, in order to maximize the usefulness of the normative background. Legislative bodies or business practices in specific sectors can be important sources for this task.

The proposed framework for contractual norms is both *adaptable* and *extensible*. By adaptable we mean that it is possible for contractual partners to change the normative background that applies to a specific contract, following the “default rules” principle as exposed above. Extensibility allows new types of e-contracts, which were not foreseen at design time, to be monitored by the normative environment, provided that such e-contracts are written in an appropriate format (a contract schema) and include their own normative core.

We have used norm defeasibility as a design tool for our normative environment, instead of its most common use as a mechanism to solve deontic conflicts. We believe that such conflicts are more likely to occur when the target actions or states of affairs are defined abstractly and need to be interpreted or translated into observable actions. In fact, legal texts usually make an intentional use of vague concepts, allowing their interpretation and application to different situations. Although this is true, in our research we have not dealt with abstraction. The hierarchical normative framework, combined with the constitutive nature of counts-as conditionals, could however be exploited to define normative relations with different degrees of abstraction. Moreover, since we are not dealing directly with permission and prohibition (in contract law, obligation is the most important deontic operator), deontic conflicts in a strict sense cannot occur.

Our approach leaves for the users of the normative environment the responsibility of checking the correctness of the contracts they establish, from a normative point of view. Formal verification techniques for e-contracts (such as those applied in the CONTRACT project [132]) could be designed that check if the combination of a normative background with contract-specific norms, as provided by our norm defeasibility model, obtains a sound contractual relation for every participant.

Contract monitoring was also addressed by our research. We have defined and implemented an infrastructure for run-time contract monitoring, based on events that feed a set of monitoring rules. Norms that apply in specific situations (following the defeasibility model already mentioned) are selected and used to prescribe normative consequences. We have shown, through experimentation, that the monitoring service correctly detects contract enactment anomalies, and reacts by applying sanctioning norms as specified in the normative framework. The examples provided show a variety of situations in which contract enactment may go wrong, and the normative environment was able to correctly respond in all cases.

A question that might be posed is whether it is plausible that this approach can be applied to monitor real-world contracts. We have followed the assumption that it is in the best interest of agents to inform the EI that they are complying with their contractual commitments. In the future, the ubiquitous use of electronic devices capturing and providing information about a vast amount of events should help on relaxing this assumption.

Another criticism that might be pointed is the centralization of the normative environment and its associated contract monitoring facility. We have based the normative environment implementation in Jess [82], a world-class rule-based inference engine. Nevertheless, when going through the creation of e-contracts in the order of thousands this centralization might not scale. The most natural way to look at this is to replicate the Electronic Institution concept: we can have multiple EIs, for instance by having each specialized into a specific business domain.

The third research question was:

- *How should e-contracts be specified in order to enable their automatic processing?*

The search for answers to this question was interleaved with the previous one. E-contract monitoring is dependent on a proper specification and representation of e-contracts.

We have looked at e-contracts from the point of view of their representation inside our structured normative environment. Here, an e-contract is translated into a *context*, including: a reference to a *parent context*, on which the norm inheritance and defeasibility model relies; the identification of the involved *agents*; a set of *contextual information* elements that specify roles, exchange values and norm parameters; and optional contract-specific *norms*, which comprise specific clauses that apply to a contract.

The e-contract schema that allows agents to use the normative environment for monitoring purposes is in Appendix E.

When formalizing norms and their obligation prescriptions, we have also looked at real-world practices from international trade regulations. We have proposed a new model for *contractual obligations*: directed obligations with livelines and deadlines. Their semantics includes a flexible approach to the temporal dimension of obligations, where counterparties are given the opportunity to participate in the monitoring process.

Finally, the last research question has lead us to research further on the *institutional* nature of an EI:

- *Which mechanisms may an Electronic Institution, and more specifically a normative environment, put in practice to enforce e-contracts?*

Prescribing contractual sanctions to autonomous entities is not enough as a mechanism for contract enforcement. It is still up to the agents whether they should fulfill sanctions. We have studied other kinds of responses an EI may include in order to enforce compliance.

Looking at the overall EI architecture, one means to incentive norm compliance is the use of trust and reputation mechanisms. While this issue is outside the scope of this thesis, the implementation of the normative environment accommodates a reporting feature that allows the environment to integrate with a Trust and Reputation service (as explained in Section 4.1).

Inside the normative environment, we have also made an initial contribution towards an adaptive model of deterrence sanctions. In this case, deterrence sanctions are regimented: agents cannot escape them, e.g. because they were required, upon entering the institution, to make a deposit that is under the control of the institution. Based on inputs from the fields of law and economics, we have designed and experimentally evaluated a model for adaptive deterrence sanctions that tries to enforce norm compliance without excessively compromising agents' willingness to establish contracts. We have studied how the model, while trying to force compliance, responds when facing different agent populations, which were characterized by a predominant level of risk tolerance and social awareness. Our experimental evaluations show that deterrence sanctions, which are materialized through fines, tend to be lower when agents are more risk-averse or more socially concerned. This means that social control is more needed when agents tend to behave more selfishly, which is in accordance with real-life experience.

In Section 9.3 we identify some lines of research related with this issue that we are currently dealing with. Before we do that, in the next section we try to take a step back and look for other potential applications of our research contributions.

9.2 Enlarging the Application Domains

We have focused our research on the application of electronic institutions, and more specifically institutional normative environments, in the field of electronic contracting. Despite this biasing domain, we believe that the ingredients we have put into our approach may be applicable in other areas.

We may characterize the class of problems that we are able to address as follows. Some joint activity between a group of agents is to be regulated by norms. This means that, although some kind of overarching goal is pursuit, we have no direct control on the behavior that agents may exhibit. The activity to be carried out may originate, at run-time, more short-termed sub-activities, which may inherit the normative specifications of the overall activity. Since agents are autonomous and may at times be more self-interested than team-players, some kind of social control is needed to enforce norm compliance. We believe that this broader perspective opens up the application domains of the research contributions we have made.

In the field of Collaborative Networked Organizations, the governance of Virtual Breeding Environments and the regulation of subsequently created Virtual Enterprises seems to match the purpose of and the developments we have made towards an institutional normative environment, particularly the context-based hierarchical normative framework. Inheriting norms from A VBE into a VE meets precisely the requirement of a fast set-up phase for VEs, by reducing the number of issues that need to be taken into account [190]. Although there is a need to generalize the kinds of regulations that are supported by the normative environment, this is a field where our contributions seem to be of value.

Some attempts have been made to apply organizational and normative notions into intrinsically cooperative collective activities such as robot soccer (e.g. [108]). In this case, using deontic notions does not hamper agent autonomy, as agents are still able to decide whether they comply with obligations, permissions or prohibitions. It is less clear, however, what sanctioning mechanisms should there be in this type of scenarios.

A research line that has been studied by many MAS researchers working with norms and institutions is that of organizational modeling. In the literature we can find some efforts on distinguishing institutions from organizations [167][101], together with their electronic counterparts [226]. We have looked into electronic institutions as computational infrastructures providing an operationalization of institutional concepts, such as norms. Looking at organizational modeling, the concept of a normative framework based on hierarchical contexts, such as we have proposed, seems to match the notions of structure and hierarchy found within organizations. Therefore, a possible extension of our work could be applied in this field.

9.3 Research Connections and Further Work

The project behind this work covers a wide range of MAS research areas, some of which we have already identified in Chapter 4. Each of these areas is very active at the present, as shows the recent initiative Agreement Technologies European Network (COST Action IC0801).

The efforts being put to develop an Electronic Institution Platform at LIACC-NIAD&R are trying to complete the circle shown in Figure 4.1. Therefore, two connections are made with the main subject of this thesis, which concerned the specification and development of an institutional normative environment for e-contracting. These can be seen as two important lines of further research, which is taking place at present:

- Automatic negotiation of e-contracts: Instead of instantiating predefined contract templates, the idea is to endow software agents with the ability to automatically negotiate specific contract clauses, taking into account trust information. Using the contract model already proposed in this thesis, we will specify contracts that are amenable to be partially negotiated in an automatic way, and negotiation protocols that address contract parameters and clauses. Advanced contract negotiation strategies (e.g. based on argumentation) need to be designed that allow agents to successfully negotiate contracts.

- Trust and reputation based on past contractual behavior: As mentioned before, the interface between the normative environment and a trust and reputation service is already implemented. However, further research is needed in this area in order to provide richer information to the computational trust and reputation model being developed [218].

Another line of research that we are addressing is related to the enforcement mechanisms that an EI may have at its disposal, continuing the initial contribution and simulation that we have included in this thesis. In particular, we are investigating on the impact trust and reputation mechanisms may have on the performance of contractual partners, and consequently on the need for other more direct enforcement mechanisms.

Appendix

Appendix A

Jess Code Listing

```
; set conflict resolution strategy to breadth, in order to generate
; notification events in the same order as Jess facts occurrence
(set-strategy breadth)

; Java-implemented user functions
(deffunction schedule-time-alert ())
(deffunction norm-env-report ())


;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
;;; TEMPLATES
;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;

;;;;;;;;;;;;;;;;;;;;;;;;
;; Contexts

(deftemplate MAIN::context
    "Context for norms"
    (slot id)
    (slot super-context (default INSTITUTIONAL-NORMS))
    (slot when)
    (multislot who) )

(deftemplate MAIN::contextual-info
    "Contextual information"
    (slot context) )

;;;;;;;;;;;;;;;;;;;;;;;;
;; Institutional reality

(deftemplate MAIN::IRE
```

```
    "Institutional reality element"
    (declare (slot-specific TRUE))
    (slot context (default INSTITUTIONAL-NORMS))
    (slot when (default-dynamic (call System currentTimeMillis)))
    (slot processing-context) )   ; used by context-handling rules

(deftemplate MAIN::start-context extends IRE
    "To indicate that a context has started" )

(deftemplate MAIN::end-context extends IRE
    "To indicate that a context has ended" )

(deftemplate MAIN::active-context
    "To indicate that a context is active"
    (slot context) )

(defrule MAIN::active-context-rule
    "Active context state"
    (logical (start-context (context ?ctx)))
    (logical (not (end-context (context ?ctx))))
    =>
    (assert (active-context (context ?ctx))) )

(deftemplate MAIN::ifact extends IRE
    "Institutional fact"
    (multislot fact) )   ; ontology for institutional-facts goes here...

(deftemplate MAIN::time extends IRE
    "Time event" )

(deftemplate MAIN::obligation extends IRE
    "Obligation"
    (slot bearer)
    (slot counterparty)
    (multislot fact)   ; ontology for institutional-facts goes here...
    (slot liveline (default 0))   ; 0 indicates no liveline
    (slot deadline) )

(deftemplate MAIN::liveline-violation extends IRE
    "Liveline violation of an obligation"
    (slot obl)
    (slot ifa) )

(deftemplate MAIN::deadline-violation extends IRE
    "Deadline violation of an obligation"
```

```
    (slot obl) )

(deftemplate MAIN::fulfillment extends IRE
    "Obligation fulfillment"
    (slot obl)
    (slot ifa) )

(deftemplate MAIN::violation extends IRE
    "Obligation violation"
    (slot obl) )

(deftemplate MAIN::denounce extends IRE
    "Temporal violation denouncement"
    (slot obl) )

;;;;;;;;;;;;;;;;;;;;;;;;
;; Brute facts

(deftemplate MAIN::bfact
    "Brute fact"
    (slot agent)
    (slot context)
    (multislot statement)   ; ontology for brute-facts goes here...
    (slot when) )

;;;;;;;;;;;;;;;;;;;;;;;;
;; Role enacting agents

(deftemplate MAIN::rea
    "Role enacting agent - agent takes a role"
    (slot agent)
    (slot role) )


;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
;;; CONSTITUTIVE RULES
;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;

(defmodule CONSTITUTIVE-RULES
    "Constitutive rules"
    (declare (auto-focus TRUE)) )

(defrule CONSTITUTIVE-RULES::payments
    "Constitutive rule for acknowledging payments"
    (bfact (agent ?a) (context ?ctx)
```

```
        (statement currency-transfered $?data)
        (when ?wh) )
    (active-context (context ?ctx))   ; check that context is active
    (rea (agent ?a) (role Bank))
    =>
    (assert (ifact (context ?ctx) (fact payment $?data) (when ?wh))) )

(defrule CONSTITUTIVE-RULES::deliveries
    "Constitutive rule for acknowledging deliveries"
    (bfact (agent ?a) (context ?ctx )
        (statement delivered $?data)
        (when ?wh) )
    (active-context (context ?ctx))   ; check that context is active
    (rea (agent ?a) (role DeliveryTracker))
    =>
    (assert (ifact (context ?ctx) (fact delivery $?data) (when ?wh))) )

(defrule CONSTITUTIVE-RULES::msg-deliveries
    "Constitutive rule for acknowledging msg-deliveries"
    (bfact (agent ?a) (context ?ctx)
        (statement msg-delivered $?data)
        (when ?wh) )
    (active-context (context ?ctx))   ; check that context is active
    (rea (agent ?a) (role Messenger))
    =>
    (assert (ifact (context ?ctx) (fact msg-delivery $?data) (when ?wh))) )

(defrule CONSTITUTIVE-RULES::denounces
    "Acknowledging denounces"
    (ifact (context ?ctx)
        (fact msg-delivery ref ? from ?fr to ?to msg denounce $?fact)
        (when ?wh) )
    ?obl <- (obligation (context ?ctx) (bearer ?to) (counterparty ?fr)
                (fact $?fact))
    =>
    (assert (denounce (context ?ctx) (obl ?obl) (when ?wh))) )

(defrule CONSTITUTIVE-RULES::msg-types
    "Acknowledging a message type (orders, ...)"
    (ifact (context ?ctx)
        (fact msg-delivery ref ?ref from ?fr to ?to
                           msg ?msg-type $?msg-contents)
        (when ?wh))
    =>
    (assert (ifact (context ?ctx)
```

```
                (fact ?msg-type ref ?ref from ?fr to ?to $?msg-contents)
                (when ?wh) ) ) )


;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
;;; INSTITUTIONAL RULES
;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;

(defmodule INSTITUTIONAL-RULES
    "Institutional rules"
    (declare (auto-focus TRUE)) )

;;;;;;;;;;;;;;;;;;;;;;;;
;; contract starting

(defrule INSTITUTIONAL-RULES::schedule-start-time-alert
    "Schedule a time event for the start of a context's life"
    (context (id ?ctx) (when ?w))
    =>
    (schedule-time-alert ?ctx ?w) )

(defrule INSTITUTIONAL-RULES::context-start
    "Signal the start of a context"
    (context (id ?ctx) (when ?w))
    (time (context ?ctx) (when ?w))
    =>
    (assert (start-context (context ?ctx) (when ?w))) )

;;;;;;;;;;;;;;;;;;;;;;;;
;; time events for directed obligations with livelines and deadlines

(defrule INSTITUTIONAL-RULES::schedule-liveline-time-alert
    "Schedule a time event at an obligation's liveline"
    ?obl <- (obligation {liveline > 0})   ; only if liveline is defined
    =>
    (schedule-time-alert ?obl.context ?obl.liveline) )

(defrule INSTITUTIONAL-RULES::schedule-deadline-time-alert
    "Schedule a time event at an obligation's deadline"
    ?obl <- (obligation)
    =>
    (schedule-time-alert ?obl.context ?obl.deadline) )

;;;;;;;;;;;;;;;;;;;;;;;;
;; semantics of directed obligations with livelines and deadlines
```

```
(defrule INSTITUTIONAL-RULES::detect-liveline-violation
    "Detect a livelive violation of an obligation"
    ?obl <- (obligation (context ?c) (fact $?f) (liveline ?liveline))
    ?ifa <- (ifact (context ?c) (fact $?f) {when < ?liveline})
    =>
    (assert (liveline-violation (context ?c) (when ?ifa.when)
                (obl ?obl) (ifa ?ifa))) )

(defrule INSTITUTIONAL-RULES::detect-early-fulfillment
    "Detect an early fulfillment of an obligation, at the liveline"
    ?lviol <- (liveline-violation (context ?c) (obl ?obl))
    ?obl <- (obligation (context ?c) (liveline ?liveline))
    ; livelive before denouncement
    (time (context ?c) (when ?liveline))
    (not (denounce (context ?c) (obl ?obl) {when <= ?liveline}))
    =>
    (assert (fulfillment (context ?c) (when ?liveline)
                (obl ?obl) (ifa ?lviol.ifa))) )

(defrule INSTITUTIONAL-RULES::detect-violation-before-liveline
    "Detect the violation of an obligation before the liveline"
    ?lviol <- (liveline-violation (context ?c) (obl ?obl))
    ?obl <- (obligation (context ?c) (liveline ?liveline))
    ; denouncement before liveline
    ?den <- (denounce (context ?c) (obl ?obl) {when < ?liveline})
    =>
    (assert (violation (context ?c) (when ?den.when) (obl ?obl))) )

(defrule INSTITUTIONAL-RULES::detect-fulfillment
    "Detect the fulfillment of an obligation"
    ?obl <- (obligation (context ?c) (fact $?f)
                (liveline ?liveline) (deadline ?deadline))
    ?ifa <- (ifact (context ?c) (fact $?f)
                {when >= ?liveline && when <= ?deadline})
    =>
    (assert (fulfillment (context ?c) (when (max ?ifa.when ?obl.when))
                (obl ?obl) (ifa ?ifa) ) ) )

(defrule INSTITUTIONAL-RULES::detect-deadline-violation
    "Detect a deadline violation of an obligation"
    ?obl <- (obligation (context ?c) (fact $?f) (deadline ?deadline))
    (time (context ?c) (when ?deadline))
    (not (ifact (context ?c) (fact $?f) {when <= ?deadline}))
    =>
```

```
    (assert (deadline-violation (context ?c) (when ?deadline)
                (obl ?obl))) )

(defrule INSTITUTIONAL-RULES::detect-belated-fulfillment
    "Detect a belated fulfillment of an obligation"
    ?dviol <- (deadline-violation (context ?c) (obl ?obl))
    ?obl <- (obligation (context ?c) (fact $?f))
    ; ifact before denouncement
    ?ifa <- (ifact (context ?c) (fact $?f) (when ?when))
    (not (denounce (context ?c) (obl ?obl) {when <= ?when}))
    =>
    (assert (fulfillment (context ?c) (when ?when)
                (obl ?obl) (ifa ?ifa))) )

(defrule INSTITUTIONAL-RULES::detect-violation-after-deadline
    "Detect the violation of an obligation after the deadline"
    ?dviol <- (deadline-violation (context ?c) (obl ?obl))
    ?obl <- (obligation (context ?c) (fact $?f))
    ; denouncement before ifact
    (denounce (context ?c) (obl ?obl) (when ?when))
    (not (ifact (context ?c) (fact $?f) {when < ?when}))
    =>
    (assert (violation (context ?c) (when ?when) (obl ?obl))) )

;;;;;;;;;;;;;;;;;;;;;;;;
;; reports

(defrule INSTITUTIONAL-RULES::new-contract-report
    "Report a new contract"
    ?ctxDef <- (context)
    =>
    (norm-env-report ?ctxDef) )

(defrule INSTITUTIONAL-RULES::ire-report
    "Report an IRE"
    ?ire <- (IRE)
    =>
    (norm-env-report ?ire) )


;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
;;; PROCESSING-CONTEXT MANAGEMENT RULES
;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;

(defrule MAIN::new-ire-processing-context
```

```
    "When a new IRE is created, set its processing-context"
    ?ire <- (IRE (context ?ctx) (processing-context nil))
    (active-context (context ?ctx))   ; check that context is active
    =>
    (modify ?ire (processing-context ?ctx)) )

(defrule MAIN::update-ire-processing-context
    "Update IRE processing-context"
    (declare (salience -100))   ; make this rule fire last (after norms)
    ?ire <- (IRE (processing-context ?pc))
    (context (id ?pc) (super-context ?sc))
    =>
    (modify ?ire (processing-context ?sc)) )

(defrule MAIN::finish-ire-processing-context
    "Finish IRE processing-context"
    (declare (salience -100))   ; make this rule fire last (after norms)
    ?ire <- (IRE (processing-context INSTITUTIONAL-NORMS))
    =>
    (modify ?ire (processing-context VOID)) )


;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
;;; INSTITUTIONAL NORMS (TOP LEVEL "DEFAULT RULES")
;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;

(deftemplate MAIN::norm-fired-on
    "To register the state fractions (IREs) on which norms fire"
    (multislot ires)   ; the IRE's on which a norm fired
    (slot context) )   ; the context of the firing norm

;;;;;;;;;;;;;;;;;;;;;;;;
;; INSTITUTIONAL-NORMS module definition

(defmodule INSTITUTIONAL-NORMS
    "Institutional norms (default rules at top institutional context)"
    (declare (auto-focus TRUE)) )


;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
;;; MISCELANEOUS FUNCTIONS AND QUERIES
;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;

; deffunction equal-sets? - used by norms
(deffunction equal-sets? (?list1 ?list2)
```

```
    "Test if two lists are sets with the same elements, in any order"
    (return (= (length$ ?list1)
               (length$ ?list2)
               (length$ (union$ ?list1 ?list2)))) )

; template and fact for scheduling time events
; at norm's pattern matching phase
(deftemplate time-scheduler
    (declare (ordered TRUE)))
(deffacts time-scheduler-fact
    (time-scheduler _) )


;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
;;; RESET
;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;

(reset)
```

Appendix B

Contract of Sale

```
(deftemplate MAIN::contract-of-sale extends contract
    "A contract of sale between a seller and a buyer" )

(deftemplate MAIN::contract-of-sale-data extends contextual-info
    "Data for contract of sale"
    (slot seller)
    (slot buyer)
    (slot product)
    (slot quantity (default 1))
    (slot unit-price (default 0.0))
    (slot delivery-rel-liveline (default 0))
    (slot delivery-rel-deadline (default 5000))
    (slot payment-rel-deadline (default 30000)) )

(defrule INSTITUTIONAL-NORMS::CoS_delivery
    "Seller obligation to deliver the product"
    (contract-of-sale (id ?ctx))
    (contract-of-sale-data (context ?ctx) (seller ?s) (buyer ?b)
        (product ?p) (quantity ?q)
        (delivery-rel-liveline ?drl) (delivery-rel-deadline ?drd) )
    (start-context (context ?ctx) (when ?w))
    =>
    (assert (obligation (context ?ctx) (bearer ?s) (counterparty ?b)
                (liveline (+ ?w ?drl)) (deadline (+ ?w ?drd))
                (fact delivery ref ?ctx from ?s to ?b
                               product ?p quantity ?q) ) ) )

(defrule INSTITUTIONAL-NORMS::CoS_payment
    "After delivery, buyer obligation to pay"
    (contract-of-sale (id ?ctx))
    (contract-of-sale-data (context ?ctx) (seller ?s) (buyer ?b)
```

```
        (quantity ?q) (unit-price ?up) (payment-rel-deadline ?prd) )
    (fulfillment (context ?ctx) (obl ?obl) (when ?w))
    ?obl <- (obligation (context ?ctx) (bearer ?s) (counterparty ?b)
                (fact delivery $?) )
    =>
    (assert (obligation (context ?ctx) (bearer ?b) (counterparty ?s)
                (deadline (+ ?w ?prd))
                (fact payment ref ?ctx from ?b to ?s
                              amount (* ?q ?up)) ) ) )

(defrule INSTITUTIONAL-NORMS::CoS_no-delivery
    "Delivery violation leads to contract end"
    (contract-of-sale (id ?ctx))
    (violation (context ?ctx) (obl ?obl) (when ?w))
    ?obl <- (obligation (context ?ctx) (fact delivery $?) )
    =>
    (assert (end-context (context ?ctx) (when ?w))) )

(defrule INSTITUTIONAL-NORMS::CoS_no-payment
    "Payment deadline violation leads to interest"
    (contract-of-sale (id ?ctx))
    (contract-of-sale-data (context ?ctx) (seller ?s) (buyer ?b)
        (payment-rel-deadline ?prd) )
    (deadline-violation (context ?ctx) (obl ?obl) (when ?w))
    ?obl <- (obligation (context ?ctx) (bearer ?b) (counterparty ?s)
                (fact payment ref ?ctx $? amount ?a) )
    =>
    (assert (obligation (context ?ctx) (bearer ?b) (counterparty ?s)
                (deadline (+ ?w ?prd))
                (fact payment ref interest from ?b to ?s
                              amount (* ?a 0.1)) ) ) )

(defrule INSTITUTIONAL-NORMS::CoS_end-of-contract
    "Delivery and payments done lead to end of contract"
    (contract-of-sale (id ?ctx))
    (fulfillment (context ?ctx) (obl ?obl_del) (when ?w_del))
    ?obl_del <- (obligation (context ?ctx) (fact delivery ref ?ctx $?))
    (fulfillment (context ?ctx) (obl ?obl_pay) (when ?w_pay))
    ?obl_pay <- (obligation (context ?ctx) (fact payment ref ?ref $?))
    (not (and ?obl <- (obligation (context ?ctx) (fact payment $?))
              (not (fulfillment (context ?ctx) (obl ?obl))) ) )
    =>
    (assert (end-context (context ?ctx) (when (max ?w_del ?w_pay)))) )
```

Appendix C

Contract Enactments

```
f-31   (start-context (context CoS-T12) (when 1279818137340))
f-33   (obligation (context CoS-T12) (when 1279818137347)
           (bearer ForOffice) (counterparty LIACC)
           (fact delivery ref CoS-T12 from ForOffice to LIACC
                 product computer-desk quantity 10)
           (liveline 1279818140340) (deadline 1279818142340))
f-36   (time (context CoS-T12) (when 1279818140340))
f-38 * (ifact (context CoS-T12) (when 1279818140872)
           (fact delivery ref CoS-T12 from ForOffice to LIACC
                 product computer-desk quantity 10))
f-39   (fulfillment (context CoS-T12) (when 1279818140872)
           (obl <Fact-33>) (ifa <Fact-38>))
f-40   (obligation (context CoS-T12) (when 1279818140888)
           (bearer LIACC) (counterparty ForOffice)
           (fact payment ref CoS-T12 from LIACC to ForOffice
                 amount 200.0)
           (liveline 0) (deadline 1279818170872))
f-45 * (ifact (context CoS-T12) (when 1279818158421)
           (fact payment ref CoS-T12 from LIACC to ForOffice
                 amount 200.0))
f-46   (fulfillment (context CoS-T12) (when 1279818158421)
           (obl <Fact-40>) (ifa <Fact-45>))
f-47   (end-context (context CoS-T12) (when 1279818158421))
```

Figure C.1: Typical contract of sale: a perfect enactment.

```
f-34   (start-context (context CoS-T12) (when 1279814477907))
f-36   (obligation (context CoS-T12) (when 1279814477992)
           (bearer ForOffice) (counterparty LIACC)
           (fact delivery ref CoS-T12 from ForOffice to LIACC
                 product computer-desk quantity 10)
           (liveline 1279814480907) (deadline 1279814482907))
f-39 * (ifact (context CoS-T12) (when 1279814479318)
           (fact delivery ref CoS-T12 from ForOffice to LIACC
                 product computer-desk quantity 10))
f-40   (liveline-violation (context CoS-T12) (when 1279814479318)
           (obl <Fact-36>) (ifa <Fact-39>))
f-42   (time (context CoS-T12) (when 1279814480907))
f-43   (fulfillment (context CoS-T12) (when 1279814480907)
           (obl <Fact-36>) (ifa <Fact-39>))
f-44   (obligation (context CoS-T12) (when 1279814480909)
           (bearer LIACC) (counterparty ForOffice)
           (fact payment ref CoS-T12 from LIACC to ForOffice
                 amount 200.0)
           (liveline 0) (deadline 1279814510907))
f-49 * (ifact (context CoS-T12) (when 1279814483701)
           (fact payment ref CoS-T12 from LIACC to ForOffice
                 amount 200.0))
f-50   (fulfillment (context CoS-T12) (when 1279814483701)
           (obl <Fact-44>) (ifa <Fact-49>))
f-51   (end-context (context CoS-T12) (when 1279814483701))
```

Figure C.2: Typical contract of sale: delivery liveline violation without denounce.

```
f-35   (start-context (context CoS-T12) (when 1279818981791))
f-37   (obligation (context CoS-T12) (when 1279818982146)
           (bearer ForOffice) (counterparty LIACC)
           (fact delivery ref CoS-T12 from ForOffice to LIACC
                 product computer-desk quantity 10)
           (liveline 1279818984791) (deadline 1279818986791))
f-40   (time (context CoS-T12) (when 1279818984791))
f-42   (time (context CoS-T12) (when 1279818986791))
f-43   (deadline-violation (context CoS-T12) (when 1279818986791)
           (obl <Fact-37>))
f-45 * (ifact (context CoS-T12) (when 1279818988248)
           (fact delivery ref CoS-T12 from ForOffice to LIACC
                 product computer-desk quantity 10))
f-46   (fulfillment (context CoS-T12) (when 1279818988248)
           (obl <Fact-37>) (ifa <Fact-45>))
f-47   (obligation (context CoS-T12) (when 1279818988264)
           (bearer LIACC) (counterparty ForOffice)
           (fact payment ref CoS-T12 from LIACC to ForOffice
                 amount 200.0)
           (liveline 0) (deadline 1279819018248))
f-50 * (ifact (context CoS-T12) (when 1279818995238)
           (fact payment ref CoS-T12 from LIACC to ForOffice
                 amount 200.0))
f-51   (fulfillment (context CoS-T12) (when 1279818995238)
           (obl <Fact-47>) (ifa <Fact-50>))
f-52   (end-context (context CoS-T12) (when 1279818995238))
```

Figure C.3: Typical contract of sale: delivery deadline violation without denounce.

```
f-31   (start-context (context CoS-T12) (when 1279820306028))
f-33   (obligation (context CoS-T12) (when 1279820306035)
           (bearer ForOffice) (counterparty LIACC)
           (fact delivery ref CoS-T12 from ForOffice to LIACC
                 product computer-desk quantity 10)
           (liveline 1279820309028) (deadline 1279820311028))
f-36   (time (context CoS-T12) (when 1279820309028))
f-38   (time (context CoS-T12) (when 1279820311028))
f-39   (deadline-violation (context CoS-T12) (when 1279820311028)
           (obl <Fact-33>))
f-41 * (ifact (context CoS-T12) (when 1279820311089)
           (fact msg-delivery ref CoS-T12 from LIACC to ForOffice
                 msg denounce delivery ref CoS-T12 from ForOffice to LIACC
                 product computer-desk quantity 10))
f-43   (denounce (context CoS-T12) (when 1279820311089) (obl <Fact-33>))
f-44   (violation (context CoS-T12) (when 1279820311089) (obl <Fact-33>))
f-45   (end-context (context CoS-T12) (when 1279820311089))
```

Figure C.4: Typical contract of sale: delivery deadline violation with denounce.

```
f-38   (start-context (context CoS-T12) (when 1279969377975))
f-40   (obligation (context CoS-T12) (when 1279969377982)
           (bearer ForOffice) (counterparty LIACC)
           (fact delivery ref CoS-T12 from ForOffice to LIACC
                 product computer-desk quantity 10)
           (liveline 1279969380975) (deadline 1279969382975))
f-43   (time (context CoS-T12) (when 1279969380975))
f-45 * (ifact (context CoS-T12) (when 1279969381804)
           (fact delivery ref CoS-T12 from ForOffice to LIACC
                 product computer-desk quantity 10))
f-46   (fulfillment (context CoS-T12) (when 1279969381804)
           (obl <Fact-40>) (ifa <Fact-45>))
f-47   (obligation (context CoS-T12) (when 1279969381819)
           (bearer LIACC) (counterparty ForOffice)
           (fact payment ref CoS-T12 from LIACC to ForOffice
                 amount 200.0)
           (liveline 0) (deadline 1279969411804))
f-52   (time (context CoS-T12) (when 1279969411804))
f-53   (deadline-violation (context CoS-T12) (when 1279969411804)
           (obl <Fact-47>))
f-54   (obligation (context CoS-T12) (when 1279969411816)
           (bearer LIACC) (counterparty ForOffice)
           (fact payment ref interest from LIACC to ForOffice
                 amount 20.0)
           (liveline 0) (deadline 1279969441804))
f-57 * (ifact (context CoS-T12) (when 1279969417837)
           (fact payment ref CoS-T12 from LIACC to ForOffice
                 amount 200.0))
f-58   (fulfillment (context CoS-T12) (when 1279969417837)
           (obl <Fact-47>) (ifa <Fact-57>))
f-60 * (ifact (context CoS-T12) (when 1279969438147)
           (fact payment ref interest from LIACC to ForOffice
                 amount 20.0))
f-61   (fulfillment (context CoS-T12) (when 1279969438147)
           (obl <Fact-54>) (ifa <Fact-60>))
f-62   (end-context (context CoS-T12) (when 1279969438147))
```

Figure C.5: Typical contract of sale: payment deadline violation.

```
f-40   (start-context (context CoS-A12) (when 1280227000680))
f-42   (obligation (context CoS-A12) (when 1280227000689)
           (bearer CPUStore) (counterparty LIACC)
           (fact delivery ref CoS-A12 from CPUStore to LIACC
                 product laptop-computer quantity 1)
           (liveline 1280227000680) (deadline 1280227005680))
f-46 * (ifact (context CoS-A12) (when 1280227001453)
           (fact delivery ref CoS-A12 from CPUStore to LIACC
                 product laptop-computer quantity 1))
f-47   (fulfillment (context CoS-A12) (when 1280227001453)
           (obl <Fact-42>) (ifa <Fact-46>))
f-52   (time (context CoS-A12) (when 1280227011453))
f-61   (obligation (context CoS-A12) (when 1280227011467)
           (bearer LIACC) (counterparty CPUStore)
           (fact payment ref CoS-A12 from LIACC to CPUStore
                 amount 800.0)
           (liveline 0) (deadline 1280227031453))
f-64 * (ifact (context CoS-A12) (when 1280227016007)
           (fact payment ref CoS-A12 from LIACC to CPUStore
                 amount 800.0))
f-65   (fulfillment (context CoS-A12) (when 1280227016007)
           (obl <Fact-61>) (ifa <Fact-64>))
f-66   (end-context (context CoS-A12) (when 1280227016007))
```

Figure C.6: Atypical contract of sale: a perfect enactment.

```
f-32   (start-context (context CoS-A12) (when 1280248565351))
f-34   (obligation (context CoS-A12) (when 1280248565359)
           (bearer CPUStore) (counterparty LIACC)
           (fact delivery ref CoS-A12 from CPUStore to LIACC
                 product laptop-computer quantity 1)
           (liveline 1280248565351) (deadline 1280248570351))
f-38 * (ifact (context CoS-A12) (when 1280248566779)
           (fact delivery ref CoS-A12 from CPUStore to LIACC
                 product laptop-computer quantity 1))
f-39   (fulfillment (context CoS-A12) (when 1280248566779)
           (obl <Fact-34>) (ifa <Fact-38>))
f-44 * (ifact (context CoS-A12) (when 1280248571817)
           (fact delivery ref return from LIACC to CPUStore
                 product laptop-computer quantity 1))
f-45   (end-context (context CoS-A12) (when 1280248571817))
```

Figure C.7: Atypical contract of sale: product is returned.

```
f-84   (start-context (context Cluster-12345) (when 1280414525379))
f-85   (start-context (context CA-123) (when 1280414525382))
f-88 * (ifact (context CA-123) (when 1280414533596)
           (fact order ref 1 from E3 to E1 product motherboard quantity 90))
f-89   (obligation (context CA-123) (when 1280414533596)
           (bearer E1) (counterparty E3)
           (fact accept ref 1 from E1 to E3 product motherboard quantity 90)
           (liveline 0) (deadline 1280414538596))
f-92 * (ifact (context CA-123) (when 1280414533741)
           (fact accept ref 1 from E1 to E3 product motherboard quantity 90))
f-93   (fulfillment (context CA-123) (when 1280414533741)
           (obl <Fact-89>) (ifa <Fact-92>))
f-95   (contract-of-sale (id CoS-T1464) (super-context CA-123)
           (when 1280414533741) (who E1 E3))
f-96   (contract-of-sale-data (context CoS-T1464) (seller E1) (buyer E3)
           (product motherboard) (quantity 90) (unit-price 40.0)
           (delivery-rel-liveline 0) (delivery-rel-deadline 5000)
           (payment-rel-deadline 30000))
f-99   (time (context CoS-T1464) (when 1280414533741))
f-100   (start-context (context CoS-T1464) (when 1280414533741))
f-102   (obligation (context CoS-T1464) (when 1280414533832)
            (bearer E1) (counterparty E3)
            (fact delivery ref CoS-T1464 from E1 to E3
                  product motherboard quantity 90)
            (liveline 1280414533741) (deadline 1280414538741))
f-108 * (ifact (context CoS-T1464) (when 1280414538030)
            (fact delivery ref CoS-T1464 from E1 to E3
                  product motherboard quantity 90))
f-109   (fulfillment (context CoS-T1464) (when 1280414538030)
            (obl <Fact-102>) (ifa <Fact-108>))
f-110   (obligation (context CoS-T1464) (when 1280414538049)
            (bearer E3) (counterparty E1)
            (fact payment ref CoS-T1464 from E3 to E1 amount 3600.0)
            (liveline 0) (deadline 1280414568030))
f-117 * (ifact (context CoS-T1464) (when 1280414562134)
            (fact payment ref CoS-T1464 from E3 to E1 amount 3600.0))
f-118   (fulfillment (context CoS-T1464) (when 1280414562134)
            (obl <Fact-110>) (ifa <Fact-117>))
f-119   (end-context (context CoS-T1464) (when 1280414562134))
```

Figure C.8: Cooperation agreement: order leading to contract of sale.

Appendix D

Car Insurance Scenario

D.1 Car Insurance Workflow Contract

```
(workflow-contract
    (id WfC-car-insurance)
    (when ...)
    (who policyholder agfil euro_assist lee_cs garage assessor) )

(action (context WfC-car-insurance) (name phoneClaim)
    (sender policyholder) (receiver euro_assist) )
(action (context WfC-car-insurance) (name receiveInfo)
    (sender policyholder) (receiver euro_assist) (rel-deadline 1000) )
(action (context WfC-car-insurance) (name assignGarage)
    (sender euro_assist) (receiver policyholder) (rel-deadline 1000) )
(action (context WfC-car-insurance) (name sendCar)
    (sender policyholder) (receiver garage) (rel-deadline 1000) )
(action (context WfC-car-insurance) (name estimateRepairCost)
    (sender garage) (receiver policyholder) (rel-deadline 2000) )
(action (context WfC-car-insurance) (name notifyClaim)
    (sender euro_assist) (receiver agfil) (rel-deadline 1000) )
(action (context WfC-car-insurance) (name forwardClaim)
    (sender agfil) (receiver lee_cs) (rel-deadline 1000) )
(action (context WfC-car-insurance) (name contactGarage)
    (sender lee_cs) (receiver garage) (rel-deadline 1000) )
(action (context WfC-car-insurance) (name sendClaimForm)
    (sender agfil) (receiver policyholder) (rel-deadline 2000) )
(action (context WfC-car-insurance) (name returnClaimForm)
    (sender policyholder) (receiver agfil) (rel-deadline 7000) )
(action (context WfC-car-insurance) (name sendRepairCost)
    (sender garage) (receiver lee_cs) (rel-deadline 1000) )
(action (context WfC-car-insurance) (name assignAssessor)
    (sender lee_cs) (receiver assessor) (rel-deadline 1000) )
```

```
(action (context WfC-car-insurance) (name inspectCar)
    (sender assessor) (receiver lee_cs) (rel-deadline 1000) )
(action (context WfC-car-insurance) (name sendNewRepairCost)
    (sender assessor) (receiver lee_cs) (rel-deadline 3000) )
(action (context WfC-car-insurance) (name agreeRepair)
    (sender lee_cs) (receiver garage) (rel-deadline 3000) )
(action (context WfC-car-insurance) (name repairCar)
    (sender garage) (receiver policyholder) (rel-deadline 5000) )
(action (context WfC-car-insurance) (name sendInvoice)
    (sender garage) (receiver lee_cs) (rel-deadline 10000) )
(action (context WfC-car-insurance) (name forwardInvoice)
    (sender lee_cs) (receiver agfil) (rel-deadline 6000) )
(action (context WfC-car-insurance) (name payRepairCost)
    (sender agfil) (receiver garage) (rel-deadline 30000) )

(constraint (context WfC-car-insurance)
    (action1 phoneClaim) (action2 receiveInfo) )
(constraint (context WfC-car-insurance)
    (action1 receiveInfo) (action2 assignGarage) )
(constraint (context WfC-car-insurance)
    (action1 receiveInfo) (action2 notifyClaim) )
(constraint (context WfC-car-insurance)
    (action1 assignGarage) (action2 sendCar) )
(constraint (context WfC-car-insurance)
    (action1 sendCar) (action2 estimateRepairCost) )
(constraint (context WfC-car-insurance)
    (action1 notifyClaim) (action2 forwardClaim) )
(constraint (context WfC-car-insurance)
    (action1 notifyClaim) (action2 sendClaimForm) )
(constraint (context WfC-car-insurance)
    (action1 forwardClaim) (action2 contactGarage) )
(constraint (context WfC-car-insurance)
    (action1 sendClaimForm) (action2 returnClaimForm) )
(constraint (context WfC-car-insurance)
    (action1 estimateRepairCost) (action2 sendRepairCost) )
(constraint (context WfC-car-insurance)
    (action1 contactGarage) (action2 sendRepairCost) )
(constraint (context WfC-car-insurance)
    (action1 sendRepairCost) (action2 assignAssessor) )
(constraint (context WfC-car-insurance)
    (action1 assignAssessor) (action2 inspectCar) )
(constraint (context WfC-car-insurance)
    (action1 inspectCar) (action2 sendNewRepairCost) )
(constraint (context WfC-car-insurance)
    (action1 sendNewRepairCost) (action2 agreeRepair) )
```

```
(constraint (context WfC-car-insurance)
    (action1 agreeRepair) (action2 repairCar) )
(constraint (context WfC-car-insurance)
    (action1 repairCar) (action2 sendInvoice) )
(constraint (context WfC-car-insurance)
    (action1 sendInvoice) (action2 forwardInvoice) )
(constraint (context WfC-car-insurance)
    (action1 forwardInvoice) (action2 payRepairCost) )
(constraint (context WfC-car-insurance)
    (action1 returnClaimForm) (action2 payRepairCost) )
```

D.2 A Phone Claim

```
f-126 * (ifact (context WfC-car-insurance) (when 1280700681109)
            (fact phoneClaim ref 1 from policyholder to euro_assist))
f-127   (obligation (context WfC-car-insurance) (when 1280700681171)
            (bearer policyholder) (counterparty euro_assist)
            (fact receiveInfo ref 1 from policyholder to euro_assist)
            (liveline 0) (deadline 1280700691109))
f-130 * (ifact (context WfC-car-insurance) (when 1280700684359)
            (fact receiveInfo ref 1 from policyholder to euro_assist))
f-131   (fulfillment (context WfC-car-insurance) (when 1280700684359)
            (obl <Fact-127>) (ifa <Fact-130>))
f-133   (obligation (context WfC-car-insurance) (when 1280700684406)
            (bearer euro_assist) (counterparty policyholder)
            (fact assignGarage ref 1 from euro_assist to policyholder)
            (liveline 0) (deadline 1280700694359))
f-135   (obligation (context WfC-car-insurance) (when 1280700684421)
            (bearer euro_assist) (counterparty agfil)
            (fact notifyClaim ref 1 from euro_assist to agfil)
            (liveline 0) (deadline 1280700694359))
f-137 * (ifact (context WfC-car-insurance) (when 1280700687937)
            (fact assignGarage ref 1 from euro_assist to policyholder))
f-138   (fulfillment (context WfC-car-insurance) (when 1280700687937)
            (obl <Fact-133>) (ifa <Fact-137>))
f-140   (obligation (context WfC-car-insurance) (when 1280700688000)
            (bearer policyholder) (counterparty garage)
            (fact sendCar ref 1 from policyholder to garage)
            (liveline 0) (deadline 1280700697937))
f-143 * (ifact (context WfC-car-insurance) (when 1280700689078)
            (fact notifyClaim ref 1 from euro_assist to agfil))
f-144   (fulfillment (context WfC-car-insurance) (when 1280700689078)
            (obl <Fact-135>) (ifa <Fact-143>))
f-146   (obligation (context WfC-car-insurance) (when 1280700689140)
            (bearer agfil) (counterparty lee_cs)
```

```
            (fact forwardClaim ref 1 from agfil to lee_cs)
            (liveline 0) (deadline 1280700699078))
f-148   (obligation (context WfC-car-insurance) (when 1280700689156)
            (bearer agfil) (counterparty policyholder)
            (fact sendClaimForm ref 1 from agfil to policyholder)
            (liveline 0) (deadline 1280700709078))
f-152 * (ifact (context WfC-car-insurance) (when 1280700691281)
            (fact forwardClaim ref 1 from agfil to lee_cs))
f-153   (fulfillment (context WfC-car-insurance) (when 1280700691281)
            (obl <Fact-146>) (ifa <Fact-152>))
f-155   (obligation (context WfC-car-insurance) (when 1280700691328)
            (bearer lee_cs) (counterparty garage)
            (fact contactGarage ref 1 from lee_cs to garage)
            (liveline 0) (deadline 1280700701281))
f-158 * (ifact (context WfC-car-insurance) (when 1280700693062)
            (fact sendCar ref 1 from policyholder to garage))
f-160   (fulfillment (context WfC-car-insurance) (when 1280700693062)
            (obl <Fact-140>) (ifa <Fact-158>))
f-161   (obligation (context WfC-car-insurance) (when 1280700693109)
            (bearer garage) (counterparty policyholder)
            (fact estimateRepairCost ref 1 from garage to policyholder)
            (liveline 0) (deadline 1280700713062))
f-167 * (ifact (context WfC-car-insurance) (when 1280700694890)
            (fact contactGarage ref 1 from lee_cs to garage))
f-169   (fulfillment (context WfC-car-insurance) (when 1280700694890)
            (obl <Fact-155>) (ifa <Fact-167>))
f-173 * (ifact (context WfC-car-insurance) (when 1280700698250)
            (fact sendClaimForm ref 1 from agfil to policyholder msg ))
f-174   (fulfillment (context WfC-car-insurance) (when 1280700698250)
            (obl <Fact-148>) (ifa <Fact-173>))
f-176   (obligation (context WfC-car-insurance) (when 1280700698296)
            (bearer policyholder) (counterparty agfil)
            (fact returnClaimForm ref 1 from policyholder to agfil)
            (liveline 0) (deadline 1280700768250))
f-183 * (ifact (context WfC-car-insurance) (when 1280700701984)
            (fact estimateRepairCost ref 1 from garage to policyholder))
f-185   (fulfillment (context WfC-car-insurance) (when 1280700701984)
            (obl <Fact-161>) (ifa <Fact-183>))
f-186   (obligation (context WfC-car-insurance) (when 1280700702031)
            (bearer garage) (counterparty lee_cs)
            (fact sendRepairCost ref 1 from garage to lee_cs)
            (liveline 0) (deadline 1280700711984))
f-189 * (ifact (context WfC-car-insurance) (when 1280700704093)
            (fact sendRepairCost ref 1 from garage to lee_cs))
f-190   (fulfillment (context WfC-car-insurance) (when 1280700704093)
```

```
            (obl <Fact-186>) (ifa <Fact-189>))
f-192   (obligation (context WfC-car-insurance) (when 1280700704156)
            (bearer lee_cs) (counterparty assessor)
            (fact assignAssessor ref 1 from lee_cs to assessor)
            (liveline 0) (deadline 1280700714093))
f-195 * (ifact (context WfC-car-insurance) (when 1280700708703)
            (fact assignAssessor ref 1 from lee_cs to assessor))
f-196   (fulfillment (context WfC-car-insurance) (when 1280700708703)
            (obl <Fact-192>) (ifa <Fact-195>))
f-198   (obligation (context WfC-car-insurance) (when 1280700708750)
            (bearer assessor) (counterparty lee_cs)
            (fact inspectCar ref 1 from assessor to lee_cs)
            (liveline 0) (deadline 1280700709703))
f-203 * (ifact (context WfC-car-insurance) (when 1280700709093)
            (fact inspectCar ref 1 from assessor to lee_cs))
f-204   (fulfillment (context WfC-car-insurance) (when 1280700709093)
            (obl <Fact-198>) (ifa <Fact-203>))
f-206   (obligation (context WfC-car-insurance) (when 1280700709140)
            (bearer assessor) (counterparty lee_cs)
            (fact sendNewRepairCost ref 1 from assessor to lee_cs)
            (liveline 0) (deadline 1280700739093))
f-217 * (ifact (context WfC-car-insurance) (when 1280700716937)
            (fact sendNewRepairCost ref 1 from assessor to lee_cs))
f-218   (fulfillment (context WfC-car-insurance) (when 1280700716937)
            (obl <Fact-206>) (ifa <Fact-217>))
f-220   (obligation (context WfC-car-insurance) (when 1280700716984)
            (bearer lee_cs) (counterparty garage)
            (fact agreeRepair ref 1 from lee_cs to garage)
            (liveline 0) (deadline 1280700746937))
f-223 * (ifact (context WfC-car-insurance) (when 1280700720406)
            (fact agreeRepair ref 1 from lee_cs to garage))
f-224   (fulfillment (context WfC-car-insurance) (when 1280700720406)
            (obl <Fact-220>) (ifa <Fact-223>))
f-226   (obligation (context WfC-car-insurance) (when 1280700720453)
            (bearer garage) (counterparty policyholder)
            (fact repairCar ref 1 from garage to policyholder)
            (liveline 0) (deadline 1280700770406))
f-229 * (ifact (context WfC-car-insurance) (when 1280700726281)
            (fact repairCar ref 1 from garage to policyholder))
f-230   (fulfillment (context WfC-car-insurance) (when 1280700726281)
            (obl <Fact-226>) (ifa <Fact-229>))
f-232   (obligation (context WfC-car-insurance) (when 1280700726312)
            (bearer garage) (counterparty lee_cs)
            (fact sendInvoice ref 1 from garage to lee_cs)
            (liveline 0) (deadline 1280700826281))
```

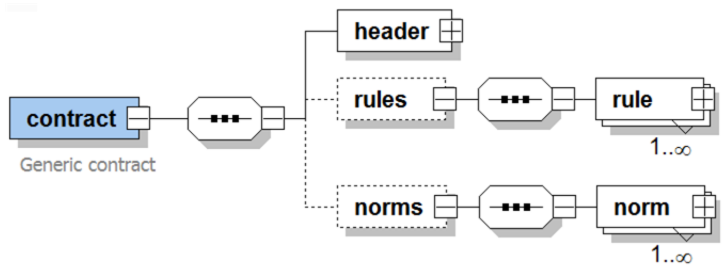
```
f-235 * (ifact (context WfC-car-insurance) (when 1280700729390)
            (fact returnClaimForm ref 1 from policyholder to agfil))
f-236   (fulfillment (context WfC-car-insurance) (when 1280700729390)
            (obl <Fact-176>) (ifa <Fact-235>))
f-241 * (ifact (context WfC-car-insurance) (when 1280700745031)
            (fact sendInvoice ref 1 from garage to lee_cs))
f-242   (fulfillment (context WfC-car-insurance) (when 1280700745031)
            (obl <Fact-232>) (ifa <Fact-241>))
f-244   (obligation (context WfC-car-insurance) (when 1280700745093)
            (bearer lee_cs) (counterparty agfil)
            (fact forwardInvoice ref 1 from lee_cs to agfil)
            (liveline 0) (deadline 1280700805031))
f-249 * (ifact (context WfC-car-insurance) (when 1280700762609)
            (fact forwardInvoice ref 1 from lee_cs to agfil))
f-250   (fulfillment (context WfC-car-insurance) (when 1280700762609)
            (obl <Fact-244>) (ifa <Fact-249>))
f-252   (obligation (context WfC-car-insurance) (when 1280700762671)
            (bearer agfil) (counterparty garage)
            (fact payRepairCost ref 1 from agfil to garage)
            (liveline 0) (deadline 1280701062609))
f-263 * (ifact (context WfC-car-insurance) (when 1280700880718)
            (fact payRepairCost ref 1 from agfil to garage))
f-265   (fulfillment (context WfC-car-insurance) (when 1280700880718)
            (obl <Fact-252>) (ifa <Fact-263>))
```

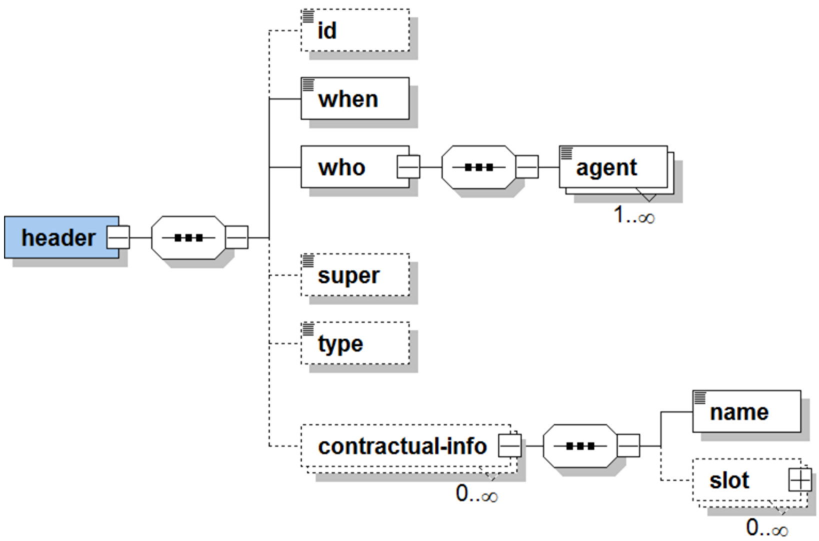
Appendix E

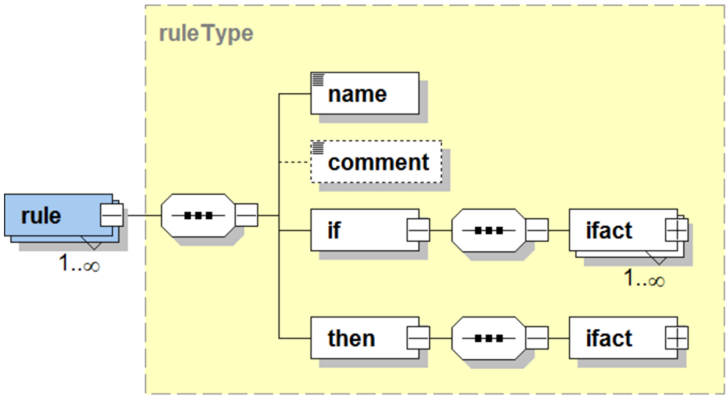
Contract XML Schema

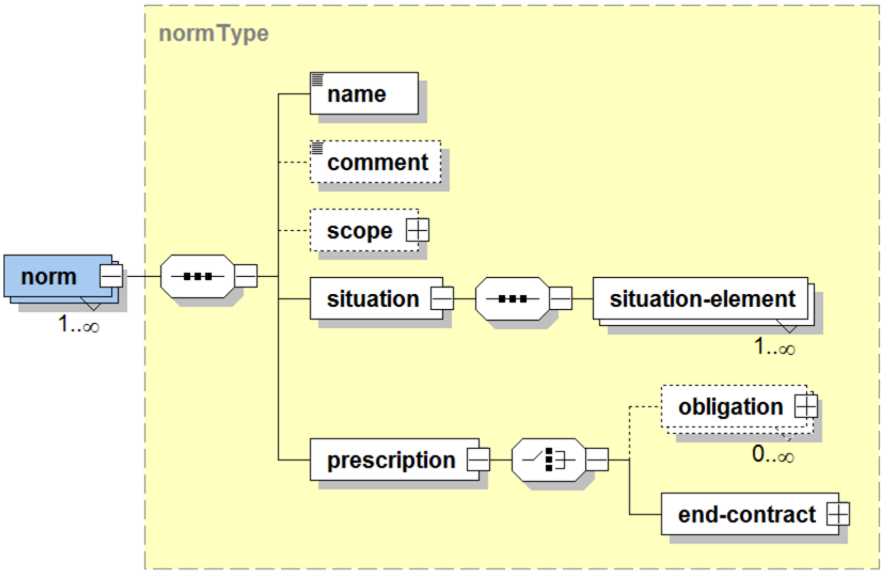
E.1 Design View Snapshots

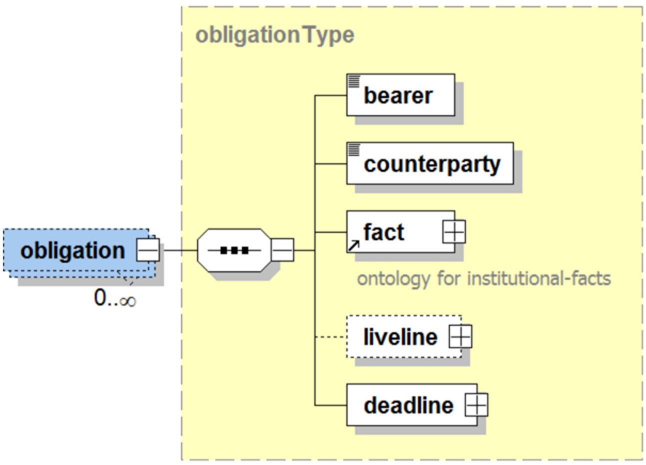
The following snapshots were taken using Altova® XMLSpy® 2006, and are included here to facilitate reading the XML schema.






ruleType
name
comment
rule
1..∞
if
ifact
1..∞
then
ifact


normType
name
comment
scope
norm
1..∞
situation
situation-element
1..∞
obligation
0..∞
prescription
end-contract


obligationType
bearer
counterparty
obligation
0..∞
fact
ontology for institutional-facts
liveline
deadline

E.2 contract.xsd

```
<?xml version="1.0" encoding="UTF-8"?>
<xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema"
           elementFormDefault="qualified" attributeFormDefault="unqualified">
 <xs:element name="contract">
  <xs:annotation>
   <xs:documentation>Generic contract</xs:documentation>
  </xs:annotation>
  <xs:complexType>
   <xs:sequence>
    <xs:element name="header">
     <xs:complexType>
      <xs:sequence>
       <xs:element name="id" type="xs:string" minOccurs="0"/>
       <xs:element name="when" type="xs:dateTime"/>
       <xs:element name="who">
        <xs:complexType>
         <xs:sequence>
          <xs:element name="agent" type="xs:string" maxOccurs="unbounded"/>
         </xs:sequence>
        </xs:complexType>
       </xs:element>
       <xs:element name="super" type="xs:string" minOccurs="0"/>
       <xs:element name="type" type="xs:string" minOccurs="0"/>
       <xs:element name="contractual-info" minOccurs="0" maxOccurs="unbounded">
        <xs:complexType>
         <xs:sequence>
          <xs:element name="name" type="xs:string"/>
          <xs:element name="slot" minOccurs="0" maxOccurs="unbounded">
           <xs:complexType>
            <xs:sequence>
             <xs:element name="name" type="xs:string"/>
             <xs:element name="value" type="xs:string"/>
            </xs:sequence>
           </xs:complexType>
          </xs:element>
         </xs:sequence>
        </xs:complexType>
       </xs:element>
      </xs:sequence>
     </xs:complexType>
    </xs:element>
    <xs:element name="rules" minOccurs="0">
     <xs:complexType>
```

```
     <xs:sequence>
      <xs:element name="rule" type="ruleType" maxOccurs="unbounded"/>
     </xs:sequence>
    </xs:complexType>
   </xs:element>
   <xs:element name="norms" minOccurs="0">
    <xs:complexType>
     <xs:sequence>
      <xs:element name="norm" type="normType" maxOccurs="unbounded"/>
     </xs:sequence>
    </xs:complexType>
   </xs:element>
  </xs:sequence>
 </xs:complexType>
</xs:element>
<xs:simpleType name="valueAttributeType">
 <xs:annotation>
  <xs:documentation>From JessML</xs:documentation>
 </xs:annotation>
 <xs:restriction base="xs:string">
  <xs:enumeration value="SYMBOL"/>
  <xs:enumeration value="STRING"/>
  <xs:enumeration value="INTEGER"/>
  <xs:enumeration value="VARIABLE"/>
  <xs:enumeration value="FACT"/>
  <xs:enumeration value="FLOAT"/>
  <xs:enumeration value="FUNCALL"/>
  <xs:enumeration value="LIST"/>
  <xs:enumeration value="JAVA_OBJECT"/>
  <xs:enumeration value="BINDING"/>
  <xs:enumeration value="MULTIVARIABLE"/>
  <xs:enumeration value="LONG"/>
  <xs:enumeration value="LAMBDA"/>
 </xs:restriction>
</xs:simpleType>
<xs:complexType name="varAllowedType" mixed="true">
 <xs:annotation>
  <xs:documentation>Similar to value element in JessML</xs:documentation>
 </xs:annotation>
 <xs:attribute name="type" type="valueAttributeType"/>
</xs:complexType>
<xs:complexType name="frameType">
 <xs:sequence>
  <xs:element name="name" type="xs:string"/>
  <xs:element name="slot" minOccurs="0" maxOccurs="unbounded">
```

```
   <xs:complexType>
    <xs:sequence>
     <xs:element name="name" type="xs:string"/>
     <xs:element name="value" type="varAllowedType"/>
    </xs:sequence>
   </xs:complexType>
  </xs:element>
 </xs:sequence>
</xs:complexType>
<xs:element name="fact" type="frameType">
 <xs:annotation>
  <xs:documentation>ontology for institutional-facts</xs:documentation>
 </xs:annotation>
</xs:element>
<xs:complexType name="expressionType">
 <xs:choice>
  <xs:sequence>
   <xs:element name="num-operator">
    <xs:simpleType>
     <xs:restriction base="xs:string">
      <xs:enumeration value="+"/>
      <xs:enumeration value="-"/>
      <xs:enumeration value="*"/>
      <xs:enumeration value="/"/>
     </xs:restriction>
    </xs:simpleType>
   </xs:element>
   <xs:element name="expression" type="expressionType" minOccurs="2"
               maxOccurs="unbounded"/>
  </xs:sequence>
  <xs:element name="operand" type="varAllowedType"/>
 </xs:choice>
</xs:complexType>
<xs:complexType name="situationElementType"/>
<xs:complexType name="andType">
 <xs:complexContent>
  <xs:extension base="situationElementType">
   <xs:sequence>
    <xs:element name="situation_element" type="situationElementType"
                maxOccurs="unbounded"/>
   </xs:sequence>
  </xs:extension>
 </xs:complexContent>
</xs:complexType>
<xs:complexType name="orType">
```

```
 <xs:complexContent>
  <xs:extension base="situationElementType">
   <xs:sequence>
    <xs:element name="situation_element" type="situationElementType"
                maxOccurs="unbounded"/>
   </xs:sequence>
  </xs:extension>
 </xs:complexContent>
</xs:complexType>
<xs:complexType name="notType">
 <xs:complexContent>
  <xs:extension base="situationElementType">
   <xs:sequence>
    <xs:element name="situation_element" type="situationElementType"/>
   </xs:sequence>
  </xs:extension>
 </xs:complexContent>
</xs:complexType>
<xs:complexType name="startContractType">
 <xs:complexContent>
  <xs:extension base="situationElementType">
   <xs:sequence>
    <xs:element name="when" type="varAllowedType"/>
   </xs:sequence>
  </xs:extension>
 </xs:complexContent>
</xs:complexType>
<xs:complexType name="contractualInfoType">
 <xs:complexContent>
  <xs:extension base="situationElementType">
   <xs:sequence>
    <xs:element name="name" type="xs:string"/>
    <xs:element name="slot" minOccurs="0" maxOccurs="unbounded">
     <xs:complexType>
      <xs:sequence>
       <xs:element name="name" type="xs:string"/>
       <xs:element name="value" type="varAllowedType"/>
      </xs:sequence>
     </xs:complexType>
    </xs:element>
   </xs:sequence>
  </xs:extension>
 </xs:complexContent>
</xs:complexType>
<xs:complexType name="ifactType">
```

```
  <xs:complexContent>
   <xs:extension base="situationElementType">
    <xs:sequence>
     <xs:element ref="fact"/>
     <xs:element name="when" type="varAllowedType"/>
    </xs:sequence>
   </xs:extension>
  </xs:complexContent>
 </xs:complexType>
 <xs:complexType name="timeType">
  <xs:complexContent>
   <xs:extension base="situationElementType">
    <xs:sequence>
     <xs:element name="when" type="varAllowedType"/>
    </xs:sequence>
   </xs:extension>
  </xs:complexContent>
 </xs:complexType>
 <xs:complexType name="livelineViolationType">
  <xs:complexContent>
   <xs:extension base="situationElementType">
    <xs:sequence>
     <xs:element name="bearer" type="varAllowedType"/>
     <xs:element name="counterparty" type="varAllowedType"/>
     <xs:element ref="fact"/>
     <xs:element name="when" type="varAllowedType"/>
    </xs:sequence>
   </xs:extension>
  </xs:complexContent>
 </xs:complexType>
 <xs:complexType name="deadlineViolationType">
  <xs:complexContent>
   <xs:extension base="situationElementType">
    <xs:sequence>
     <xs:element name="bearer" type="varAllowedType"/>
     <xs:element name="counterparty" type="varAllowedType"/>
     <xs:element ref="fact"/>
     <xs:element name="when" type="varAllowedType"/>
    </xs:sequence>
   </xs:extension>
  </xs:complexContent>
 </xs:complexType>
 <xs:complexType name="fulfillmentType">
  <xs:complexContent>
   <xs:extension base="situationElementType">
```

```
    <xs:sequence>
     <xs:element name="bearer" type="varAllowedType"/>
     <xs:element name="counterparty" type="varAllowedType"/>
     <xs:element ref="fact"/>
     <xs:element name="when" type="varAllowedType"/>
    </xs:sequence>
   </xs:extension>
  </xs:complexContent>
 </xs:complexType>
 <xs:complexType name="violationType">
  <xs:complexContent>
   <xs:extension base="situationElementType">
    <xs:sequence>
     <xs:element name="bearer" type="varAllowedType"/>
     <xs:element name="counterparty" type="varAllowedType"/>
     <xs:element ref="fact"/>
     <xs:element name="when" type="varAllowedType"/>
    </xs:sequence>
   </xs:extension>
  </xs:complexContent>
 </xs:complexType>
 <xs:complexType name="relConditionType">
  <xs:complexContent>
   <xs:extension base="situationElementType">
    <xs:sequence>
     <xs:element name="rel-operator">
      <xs:simpleType>
       <xs:restriction base="xs:string">
        <xs:enumeration value="eq"/>
        <xs:enumeration value="gt"/>
        <xs:enumeration value="ge"/>
        <xs:enumeration value="lt"/>
        <xs:enumeration value="le"/>
        <xs:enumeration value="ne"/>
       </xs:restriction>
      </xs:simpleType>
     </xs:element>
     <xs:element name="expression" type="expressionType" minOccurs="2"
                 maxOccurs="unbounded"/>
    </xs:sequence>
   </xs:extension>
  </xs:complexContent>
 </xs:complexType>
 <xs:complexType name="ruleType">
  <xs:sequence>
```

```
   <xs:element name="name" type="xs:string"/>
   <xs:element name="comment" type="xs:string" minOccurs="0"/>
   <xs:element name="if">
    <xs:complexType>
     <xs:sequence>
      <xs:element name="ifact" type="ifactType" maxOccurs="unbounded"/>
     </xs:sequence>
    </xs:complexType>
   </xs:element>
   <xs:element name="then">
    <xs:complexType>
     <xs:sequence>
      <xs:element name="ifact" type="ifactType"/>
     </xs:sequence>
    </xs:complexType>
   </xs:element>
  </xs:sequence>
 </xs:complexType>
 <xs:complexType name="normType">
  <xs:sequence>
   <xs:element name="name" type="xs:string"/>
   <xs:element name="comment" type="xs:string" minOccurs="0"/>
   <xs:element name="scope" minOccurs="0">
    <xs:complexType>
     <xs:sequence>
      <xs:element name="contract-type" type="xs:string" minOccurs="0"/>
      <xs:element name="id" type="varAllowedType"/>
     </xs:sequence>
    </xs:complexType>
   </xs:element>
   <xs:element name="situation">
    <xs:complexType>
     <xs:sequence>
      <xs:element name="situation-element" type="situationElementType"
                  maxOccurs="unbounded"/>
     </xs:sequence>
    </xs:complexType>
   </xs:element>
   <xs:element name="prescription">
    <xs:complexType>
     <xs:choice>
      <xs:element name="obligation" type="obligationType" minOccurs="0"
                  maxOccurs="unbounded"/>
      <xs:element name="end-contract" type="endContractType"/>
     </xs:choice>
```

```
    </xs:complexType>
   </xs:element>
  </xs:sequence>
 </xs:complexType>
 <xs:complexType name="obligationType">
  <xs:sequence>
   <xs:element name="bearer" type="varAllowedType"/>
   <xs:element name="counterparty" type="varAllowedType"/>
   <xs:element ref="fact"/>
   <xs:element name="liveline" type="expressionType" minOccurs="0"/>
   <xs:element name="deadline" type="expressionType"/>
  </xs:sequence>
 </xs:complexType>
 <xs:complexType name="endContractType">
  <xs:sequence>
   <xs:element name="when" type="varAllowedType"/>
  </xs:sequence>
 </xs:complexType>
</xs:schema>
```

Bibliography

[1] A. S. Abrahams and J. M. Bacon. Representing and Enforcing E-Commerce Contracts Using Occurrences. In *Fourth International Conference on Electronic Commerce Research (ICECR-4)*, pages 59–82, Dallas, Texas, USA, 2001.

[2] Z. J. Acs and L. Preston. Small and Medium-Sized Enterprises, Technology, and Globalization: Introduction to a special issue on Small and Medium-Sized Enterprises in the Global Economy. *Small Business Economics*, 9(1):1–6, 1997.

[3] H. Aldewereld. *Autonomy vs. Conformity: an Institutional Perspective on Norms and Protocols*. PhD thesis, Utrecht University, 2007.

[4] G. Andrighetto, M. Campennì, R. Conte, and M. Paolucci. On the Immergence of Norms: a Normative Agent Architecture. In *AAAI Symposium, Social and Organizational Aspects of Intelligence*, Washington DC, 2007.

[5] G. Andrighetto, R. Conte, P. Turrini, and M. Paolucci. Emergence In the Loop: Simulating the two way dynamics of norm innovation. In G. Boella, L. van der Torre, and H. Verhagen, editors, *Normative Multi-agent Systems*, volume 07122 of *Dagstuhl Seminar Proceedings*, Dagstuhl, Germany, 2007. Schloss Dagstuhl.

[6] S. Angelov. *Foundations of B2B Electronic Contracting*. PhD thesis, Technical University Eindhoven, 2006.

[7] S. Angelov and P. Grefen. An Approach to the Construction of Flexible B2B E-Contracting Processes. Technical report, University of Twente, 2002.

[8] S. Angelov and P. Grefen. The 4W framework for B2B e-contracting. *International Journal of Networking and Virtual Organisations*, 2(1):78–97, 2003.

[9] S. Angelov and P. Grefen. The Business Case for B2B E-Contracting. In M. Janssen, H. G. Sol, and R. W. Wagenaar, editors, *6th International Conference on Electronic Commerce*, pages 31–40, Delft, The Netherlands, 2004. ACM.

[10] J. L. Arcos, M. Esteva, P. Noriega, J. A. Rodríguez-Aguilar, and C. Sierra. Environment Engineering for Multiagent Systems. *Engineering Applications of Artificial Intelligence*, 18(2):191–204, 2005.

[11] A. Artikis. *Executable Specification of Open Norm-Governed Computational Systems*. PhD thesis, Imperial College London, 2003.

[12] A. Artikis, L. Kamara, J. Pitt, and M. Sergot. A Protocol for Resource Sharing in Norm-Governed Ad Hoc Networks. In J. A. Leite, A. Omicini, P. Torroni, and P. Yolum, editors, *Declarative Agent Languages and Technologies II*, Lecture Notes in Computer Science, pages 221–238. Springer, 2004.

[13] A. Artikis, J. Pitt, and M. Sergot. Animated Specifications of Computational Societies. In C. Castelfranchi and W. L. Johnson, editors, *First International Joint Conference on Autonomous Agents and Multiagent Systems*, pages 1053–1062, Bologna, Italy, 2002. ACM, New York, USA.

[14] Assembleia da República Portuguesa. Decreto-Lei nº 47344/66: Código Civil, 1966.

[15] Assembleia da República Portuguesa. Decreto-Lei nº 7/2004 sobre Comércio Electrónico, 2004.

[16] P. S. Atiyah. *An Introduction to the Law of Contract*. Clarendon Law Series. Clarendon Press, Oxford, 4th edition edition, 1989.

[17] J. L. Austin. *How to Do Things with Words*. Oxford University Press, Oxford, 1962.

[18] F. L. Bellifemine, G. Caire, and D. Greenwood. *Developing Multi-Agent Systems with JADE*. Wiley Series in Agent Technology. John Wiley & Sons, Ltd, 2007.

[19] M. B. Blake. Agent-oriented approaches to B2B interoperability. *The Knowledge Engineering Review*, 16(4):383–388, 2001.

[20] G. Boella and L. Lesmo. Deliberate Normative Agents. In R. Conte and C. Dellarocas, editors, *Social Order in Multiagent Systems*, pages 85–110. Kluwer, 2001.

[21] G. Boella and L. van der Torre. Contracts as Legal Institutions in Organizations of Autonomous Agents. In N.R. Jennings, C. Sierra, L. Sonenberg, and M. Tambe, editors, *Third International Joint Conference on Autonomous Agents and Multiagent Systems*, volume 2, pages 948–955, New York, United States, 2004. ACM Press.

[22] G. Boella and L. van der Torre. A Game-Theoretic Approach to Normative Multi-Agent Systems. In G. Boella, L. van der Torre, and H. Verhagen, editors, *Normative Multi-agent Systems*, volume 07122 of *Dagstuhl Seminar Proceedings*, Dagstuhl, Germany, 2007. Schloss Dagstuhl.

[23] G. Boella, L. van der Torre, and H. Verhagen. Introduction to normative multiagent systems. *Computational & Mathematical Organization Theory*, 12(2-3):71–79, 2006.

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

[25] R. I. Brafman and M. Tennenholtz. On Partially Controlled Multi-Agent Systems. *Artificial Intelligence Research*, 4:477–507, 1996.

[26] J. Broersen, M. Dastani, J. Hulstijn, and L. van der Torre. Goal Generation in the BOID Architecture. *Cognitive Science Quarterly*, 2(3-4):428–447, 2002.

[27] J. Broersen, F. Dignum, V. Dignum, and J.-J. Ch. Meyer. Designing a Deontic Logic of Deadlines. In A. Lomuscio and D. Nute, editors, *7th International Workshop on Deontic Logic in Computer Science*, volume 3065 of *Lecture Notes in Computer Science*, pages 43–56, Madeira, Portugal, 2004. Springer Verlag, Heidelberg, D-69121, Germany.

[28] L. M. Camarinha-Matos and H. Afsarmanesh. The Virtual Enterprise Concept. In L.M. Camarinha-Matos and H. Afsarmanesh, editors, *Infrastructures for Virtual Enterprises: Networking Industrial Enterprises*, pages 3–14. Kluwer Academic Publishers, 1999.

[29] L. M. Camarinha-Matos and H. Afsarmanesh. Elements of a base VE infrastructure. *Journal of Computers in Industry*, 51(2):139–163, 2003.

[30] L. M. Camarinha-Matos and H. Afsarmanesh. Some Basic Concepts. In L.M. Camarinha-Matos and H. Afsarmanesh, editors, *Collaborative Networked Organizations: A research agenda for emerging business models*, pages 7–10. Springer, 2004.

[31] 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.

[32] J. Carmo and A. Jones. Deontic logic and contrary-to-duties. In D. Gabbay and F. Guenthner, editors, *Handbook of Philosophical Logic*, volume 8, pages 265–343. Kluwer, 2nd edition, 2002.

[33] C. Carter, E. White, T. Hassan, M. Shelbourn, and A. Baldwin. Legal Issues of Collaborative Electronic Working in Construction. *Proceedings of the Institution of Civil Engineers – Civil Engineering*, 150(6):10–16, 2002.

[34] C. Castelfranchi. Engineering Social Order. In A. Omicini, R. Tolksdorf, and F. Zambonelli, editors, *Engineering Societies in the Agents World*, volume 1972 of *Lecture Notes in Artificial Intelligence*, pages 1–18. Springer, Berlin, Germany, 2000.

[35] C. Castelfranchi, F. Dignum, C. M. Jonker, and J. Treur. Deliberative Normative Agents: Principles and Architectures. In N. Jennings and Y. Lesperance, editors, *Intelligent Agents VI: Agent Theories, Architectures, and Languages*, volume 1757 of *Lecture Notes in Artificial Intelligence*. Springer, Orlando, Florida, USA, 2000.

[36] A. Chopra, A. Mallya, N. Desai, and M. Singh. Modeling Flexible Business Processes. In J. A. Leite, A. Omicini, P. Torroni, and P. Yolum, editors, *Declarative Agent Languages and Technologies, Second International Workshop*, pages 93–108, New York, 2004.

[37] A. Chopra and M. Singh. Nonmonotonic Commitment Machines. In F. Dignum, editor, *Advances in Agent Communication*, Lecture Notes on Artificial Intelligence, pages 183–200. Springer, 2004.

[38] 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.

[39] M. Colombetti and M. Verdicchio. An Analysis of Agent Speech Acts as Institutional Actions. In C. Castelfranchi and W. L. Johnson, editors, *First International Joint Conference on Autonomous Agents and Multiagent Systems*, volume 3, pages 1157–1164. Association for Computing Machinery, New York, NY 10036-5701, United States, Bologna, Italy, 2002.

[40] R. Confalonieri, S. Álvarez Napagao, S. Panagiotidi, J. Vázquez-Salceda, and S. Willmott. A Middleware Architecture for Building Contract-Aware

Agent-Based Services. In R. Kowalczyk, M. Huhns, M. Klusch, Z. Maamar, and Q. B. Vo, editors, *Service-Oriented Computing: Agents, Semantics, and Engineering*, Lecture Notes in Computer Science, pages 1–14. Springer, 2008.

[41] R. Conte, G. Andrighetto, and M. Campennì. On Norm Internalization. A Position Paper. In B. Edmonds and N. Gilbert, editors, *The 6th Conference of the European Social Simulation Association*, Guildford, Surrey, UK, 2009. The European Social Simulation Association.

[42] R. Conte and C. Castelfranchi. *Cognitive and Social Action.* UCL Press, London, 1995.

[43] R. Conte and C. Castelfranchi. From conventions to prescriptions. Towards an integrated view of norms. *Artificial Intelligence and Law*, 7:323–340, 1999.

[44] R. Conte, C. Castelfranchi, and F. Dignum. Autonomous norm acceptance. In J. P. Müller, M. P. Singh, and A. S. Rao, editors, *Intelligent Agents V. Agents Theories, Architectures, and Languages*, volume 1555 of *Lecture Notes in Artificial Intelligence*, pages 99–112. Springer, Paris, France, 1999.

[45] R. Conte, R. Falcone, and G. Sartor. Introduction: Agents and Norms: How to fill the gap? *Artificial Intelligence and Law*, 7(1):1–15, 1999.

[46] S. Cranefield. A Rule Language for Modelling and Monitoring Social Expectations in Multi-agent Systems. In O. Boissier, J. Padget, V. Dignum, G. Lindemann, E. Matson, S. Ossowski, J. S. Sichman, and J. Vázquez-Salceda, editors, *Coordination, Organizations, Institutions, and Norms in Multi-Agent Systems*, Lecture Notes in Artificial Intelligence, pages 246–258. Springer, 2006.

[47] S. Cranefield. Modelling and Monitoring Social Expectations in Multi-agent Systems. 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 308–321. Springer, 2007.

[48] R. Craswell. Contract Law: General Theories. In B. Bouckaert and G. De Geest, editors, *Encyclopedia of Law and Economics*, volume Volume III: The Regulation of Contracts, pages 1–24. Edward Elgar, Cheltenham, UK, 2000.

[49] A. Dan, D. Dias, T. Nguyen, M. Sachs, H. Shaikh, R. King, and S. Duri. The Coyote Project: Framework for Multi-party E-Commerce. In C. Nikolaou and C. Stephanidis, editors, *Second European Conference on Research and*

Advanced Technology for Digital Libraries, pages 873–889, Heraklion, Crete, Greece, 1998. Springer.

[50] A. Daskalopulu. *Logic-Based Tools for the Analysis and Representation of Legal Contracts*. PhD thesis, University of London, 1999.

[51] A. Daskalopulu. Modelling Legal Contracts as Processes. In *11th International Conference and Workshop on Database and Expert Systems Applications*, volume Legal Information Systems Applications, pages 1074–1079. IEEE CS Press, 2000.

[52] A. Daskalopulu, T. Dimitrakos, and T. Maibaum. Evidence-Based Electronic Contract Performance Monitoring. *Group Decision and Negotiation*, 11(6):469–485, 2002.

[53] A. Daskalopulu and T. Maibaum. Towards Electronic Contract Performance. In *12th International Conference and Workshop on Database and Expert Systems Applications*, pages 771–777. IEEE C. S. Press, 2001.

[54] A. Daskalopulu and M. Sergot. The Representation of Legal Contracts. *AI and Society*, 11(1, 2):6–17, 1997.

[55] H. Davulcu, M. Kifer, L. R. Pokorny, C. R. Ramakrishnan, I. V. Ramakrishnan, and S. Dawson. Modeling and Analysis of Interactions in Virtual Enterprises. In *Ninth International Workshop on Research Issues on Data Engineering: Information Technology for Virtual Enterprises*, pages 12–18. IEEE Computer Society, 1999.

[56] T. De Bree and V. Furundarena. Mediating and Monitoring Electronic Commerce. In B. Stanford-Smith and P.T. Kidd, editors, *E-business: Key Issues, Applications and Technologies*, pages 232–238. IOS Press, 2000.

[57] C. Dellarocas. Contractual Agent Societies: Negotiated shared context and social control in open multi-agent systems. In *Workshop on Norms and Institutions in Multi-Agent Systems*, Barcelona, Spain, 2000.

[58] N. Desai, N. C. Narendra, and M. P. Singh. Checking Correctness of Business Contracts via Commitments. In Padgham, Parkes, Müller, and Parsons, editors, *7th Int. Conf. on Autonomous Agents and Multiagent Systems (AAMAS 2008)*, pages 787–794, Estoril, Portugal, 2008. IFAAMAS. 1402333 787-794.

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

[60] F. Dignum. Agents, markets, institutions and protocols. In F. Dignum and C. Sierra, editors, *Agent Mediated Electronic Commerce: The European Agentlink perspective*, volume LNAI 1991, pages 98–114. Springer, 2001.

[61] F. Dignum. Abstract Norms and Electronic Institutions. In *International Workshop on Regulated Agent-Based Social Systems: Theories and Applications (RASTA'02)*, Bologna, Italy, 2002.

[62] F. Dignum, J. Broersen, V. Dignum, and J.-J. Meyer. Meeting the deadline: Why, when and how. In *3rd Conference on Formal Aspects of Agent-Based Systems (FAABS III)*, volume 3228 of *Lecture Notes in Artificial Intelligence*, pages 30–40, Greenbelt, MD, United States, 2004. Springer Verlag.

[63] F. Dignum and R. Kuiper. Specifying Deadlines with Dense Time Using Deontic and Temporal Logic. *International Journal of Electronic Commerce*, 3(2):67–86, 1998.

[64] V. Dignum. *A Model for Organizational Interaction: Based on Agents, Founded in Logic*. PhD thesis, Utrecht University, 2004.

[65] V. Dignum and F. Dignum. Modelling agent societies: co-ordination frameworks and institutions. In P. Brazdil and A. Jorge, editors, *Progress in Artificial Intelligence: Knowledge Extraction, Multi-agent Systems, Logic Programming, and Constraint Solving*, volume 2258 of *Lecture Notes in Artificial Intelligence*, pages 191–204. Springer, 2001.

[66] V. Dignum, J.-J. C. Meyer, F. Dignum, and H. Weigand. Formal Specification of Interaction in Agent Societies. In M. Hinchey, J. Rash, W. Truszkowski, C. Rouff, and D. Gordon-Spears, editors, *Formal Approaches to Agent-Based Systems*, volume 2699 of *Lecture Notes in Artificial Intelligence*, pages 37–52. Springer Verlag, Heidelberg, Germany, Greenbelt, MD, United States, 2003.

[67] E. A. Emerson. Temporal and Modal Logic. In J. van Leeuwen, editor, *Handbook of Theoretical Computer Science*, volume Volume B: Formal Models and Semantics, pages 995–1072. North-Holland Pub. Co./MIT Press, 1990.

[68] M. Esteva. *Electronic Institutions: from specification to development*. PhD thesis, Technical University of Catalonia, 2003.

[69] M. Esteva, D. de la Cruz, and C. Sierra. ISLANDER: an electronic institutions editor. In C. Castelfranchi and W.L. Johnson, editors, *First International Joint Conference on Autonomous Agents and Multiagent Systems*, pages 1045–1052, Bologna, Italy, 2002. ACM Press.

[70] M. Esteva, J. Padget, and C. Sierra. Formalizing a language for institutions and norms. In J.-J. Meyer and M. Tambe, editors, *Intelligent Agents VIII*, Lecture Notes in Artificial Intelligence, pages 348–366. Springer, 2002.

[71] M. Esteva, J. A. Rodríguez-Aguilar, C. Sierra, P. Garcia, and J. L. Arcos. On the formal specifications of electronic institutions. In F. Dignum and C. Sierra, editors, *Agent-mediated Electronic commerce: The European AgentLink Perspective*, volume 1991 of *Lecture Notes in Artificial Intelligence*, pages 126–147. Springer, 2001.

[72] 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.

[73] P. Evans and T. S. Wurster. *Blown to Bits: How the New Economics of Information Transforms Strategy.* Harvard Business School Press, 1999.

[74] J. Ferber and J.-P. Müller. Influences and Reaction: a Model of Situated Multiagent Systems. In *Second International Conference on Multiagent Systems*, pages 72–79, Japan, 1996. AAAI Press.

[75] T. Finin, R. Fritzson, D. McKay, and R. McEntire. KQML as an Agent Communication Language. In *Third International Conference on Information and Knowledge Management (CIKM'94)*, pages 456–463. ACM Press, 1994.

[76] K. Fischer, J. P. Müller, I. Heimig, and A.-W. Scheer. Intelligent Agents in Virtual Enterprises. In *First International Conference on Practical Applications of Intelligent Agents and Multi-Agent Technology (PAAM'96)*, pages 205–223, London, 1996.

[77] N. Fornara and M. Colombetti. Defining Interaction Protocols using a Commitment-based Agent Communication Language. In *Second International Joint Conference on Autonomous Agents and Multiagent Systems*, pages 520–527, Melbourne, Australia, 2003. ACM, New York, USA.

[78] 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.

[79] N. Fornara, F. Viganò, and M. Colombetti. Agent Communication and Institutional Reality. In R. M. van Eijk, M.-P. Huget, and F. Dignum, editors, *Agent Communication: International Workshop on Agent Communication*,

volume 3396 of *Lecture Notes in Computer Science*, pages 1–17, New York, NY, United States, 2005. Springer Verlag, Heidelberg, D-69121, Germany.

[80] N. Fornara, F. Viganò, M. Verdicchio, and M. Colombetti. Artificial Institutions: A Model of Institutional Reality for Open Multiagent Systems. *Artificial Intelligence and Law*, 16(1):89–105, 2008.

[81] Foundation for Intelligent Physical Agents. FIPA ACL Message Structure Specification.

[82] E. Friedman-Hill. *Jess in Action.* Manning Publications Co., 2003.

[83] A. García-Camino. Ignoring, Forcing and Expecting Simultaneous Events in Electronic Institutions. In J.S. Sichman, J. Padget, S. Ossowski, and P. Noriega, editors, *Coordination, Organizations, Institutions, and Norms in Agent Systems III*, LNAI 4870, pages 15–26. Springer, 2008.

[84] A. García-Camino. *Normative Regulation of Open Multi-agent Systems.* PhD thesis, Universitat Autònoma de Barcelona, 2008.

[85] A. García-Camino, P. Noriega, and J. A. Rodríguez-Aguilar. Implementing Norms in Electronic Institutions. In F. Dignum, V. Dignum, S. Koenig, S. Kraus, M.P. Singh, and M. Wooldridge, editors, *Fourth International Joint Conference on Autonomous Agents and Multiagent Systems*, volume 2, pages 667–673, Utrecht, The Netherlands, 2005. ACM Press.

[86] A. García-Camino, P. Noriega, and J. A. Rodríguez-Aguilar. An Algorithm for Conflict Resolution in Regulated Compound Activities. In G. M. P. O'Hare, A. Ricci, M. J. O'Grady, and O. Dikenelli, editors, *Engineering Societies in the Agents World VII*, volume 4457 of *Lecture Notes in Computer Science*, pages 193–208. Springer, 2007.

[87] A. García-Camino, J. A. Rodríguez-Aguilar, C. Sierra, and W. Vasconcelos. Norm-Oriented Programming of Electronic Institutions: A Rule-Based Approach. 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 177–193. Springer, 2007.

[88] N. Garoupa. The Theory of Optimal Law Enforcement. *Journal of Economic Surveys*, 11(3):267–295, 1997.

[89] G. K. Giannikis and A. Daskalopulu. Defeasible Reasoning with e-Contracts. In *IEEE/WIC/ACM International Conference on Intelligent Agent Technology*, pages 690–694, 2006.

[90] J. Giarratano and G. Riley. *Expert Systems: Principles and Programming.* PWS Publishing Company, third edition edition, 1998.

[91] H. Gimpel, N. R. Jennings, G. E. Kersten, A. Ockenfels, and C. Weinhardt. Market Engineering: A Research Agenda. In *Negotiation, Auctions, and Market Engineering*, volume 2 of *Lecture Notes in Business Information Processing*, pages 1–15. Springer, 2006.

[92] M. Gisler, K. Stanoevska-Slabeva, and M. Greunz. Legal Aspects of Electronic Contracts. In H. Ludwig, Y. Hoffner, C. Bussler, and M. Bichler, editors, *CAISE*00 Workshop on Infrastructure for Dynamic Business-to-Business Service Outsourcing*, volume 30, Stockholm, Sweden, 2000. CEUR Workshop Proceedings.

[93] A. Goodchild, C. Herring, and Z. Milosevic. Business Contracts for B2B. In H. Ludwig, Y. Hoffner, C. Bussler, and M. Bichler, editors, *CAISE*00 Workshop on Infrastructure for Dynamic Business-to-Business Service Outsourcing*, volume 30, pages 63–74, Stockholm, Sweden, 2000. CEUR Workshop Proceedings.

[94] G. Governatori. Representing Business Contracts in RuleML. *International Journal of Cooperative Information Systems*, 14(2-3):181–216, 2005.

[95] 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.

[96] M. Greunz, B. Schopp, and K. Stanoevska-Slabeva. Supporting Market Transactions through XML Contracting Containers. In *Sixth Americas Conference on Information Systems (AMCISS 2000)*, Long Beach, CA, 2000.

[97] F. Griffel, M. Boger, H. Weinreich, W. Lamersdorf, and M. Merz. Electronic Contracting with COSMOS – How to Establish, Negotiate and Execute Electronic Contracts on the Internet. In C. Kobryn, C. Atkinson, and Z. Milosevic, editors, *2nd International Enterprise Distributed Object Computing Workshop (EDOC'98)*, pages 46–55, San Diego, California, 1998. IEEE.

[98] B. N. Grosof. Representing e-commerce rules via situated courteous logic programs in RuleML. *Electronic Commerce Research and Applications*, 3(1):2–20, 2004.

[99] B. N. Grosof, Y. Labrou, and H. Y. Chan. A Declarative Approach to Business Rules in Contracts: Courteous Logic Programs in XML. In *ACM Conference on Electronic Commerce*, pages 68–77, 1999.

[100] B. N. Grosof and T. C. Poon. Representing Agent Contracts With Exceptions using XML Rules, Ontologies, and Process Descriptions. In *International Workshop on Rule Markup Languages for Business Rules on the Semantic Web*, Sardinia, Italy, 2002.

[101] D. Grossi. *Designing Invisible Handcuffs. Formal Investigations in Institutions and Organizations for Multi-agent Systems.* Phd, Utrecht University, 2007.

[102] 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.

[103] D. Grossi and F. Dignum. From abstract to concrete norms in agent institutions. In *Proceedings of the Third NASA Workshop on Formal Approaches to Agent-Based Systems FAABS III*, volume 3228 of *Lecture Notes in Artificial Intelligence*, pages 12–29, Greenbelt, MD, United States, 2004. Springer Verlag.

[104] D. Grossi, F. Dignum, and J.-J. Ch. Meyer. Contextual Taxonomies. In J. Leite and P. Torroni, editors, *Computational Logic in Multi-Agent Systems, 5th International Workshop (CLIMA V)*, volume 3487 of *LNAI*, pages 33–51. Springer, 2005.

[105] P. Haddawy and S. Hanks. Utility Models for Goal-Directed, Decision-Theoretic Planners. *Computational Intelligence*, 14(3):392–429, 1998.

[106] M. Hannoun, O. Boissier, J. S. Sichman, and C. Sayettat. MOISE: An Organizational Model for Multi-agent Systems. In M. C. Monard and J. S. Sichman, editors, *Advances in Artificial Intelligence – International Joint Conference 7th Ibero-American Conference on AI, 15th Brazilian Symposium on AI (IBERAMIA-SBIA)*, LNAI, pages 156–165, Atibaia, SP, Brazil, 2000. Springer.

[107] J. F. Hübner, J. S. Sichman, and O. Boissier. A Model for the Structural, Functional, and Deontic Specification of Organizations in Multiagent Systems. In G. Bittencourt and G. L. Ramalho, editors, *Advances in Artificial Intelligence – 16th Brazilian Symposium on Artificial Intelligence*, pages 439–448, Porto de Galinhas/Recife, Brazil, 2002. Springer.

[108] J. F. Hübner, J. S. Sichman, and O. Boissier. Developing organised multiagent systems using the MOISE+ model: Programming issues at the system and agent levels. *International Journal of Agent-Oriented Software Engineering*, 1(3/4):370–395, 2007.

[109] M. He, N. R. Jennings, and H.-F. Leung. On agent-mediated electronic commerce. *IEEE Transactions on Knowledge and Data Engineering*, 15(4):985–1003, 2003.

[110] H. Herrestad and C. Krogh. Obligations Directed from Bearers to Counterparties. In *Proceedings of the 5th international conference on Artificial intelligence and law*, pages 210–218, College Park, Maryland, United States, 1995. ACM.

[111] C. Hewitt. Open Information Systems Semantics for Distributed Artificial Intelligence. *Artificial Intelligence*, 47:79–106, 1991.

[112] Y. Hoffner, S. Field, P. Grefen, and H. Ludwig. Contract-Driven Creation and Operation of Virtual Enterprises. *International Journal of Computer and Telecommunications Networking*, 37(2):111–136, 2001.

[113] Y. Hoffner, H. Ludwig, P. Grefen, and K. Aberer. CrossFlow: integrating workflow management and electronic commerce. *ACM SIGecom Exchanges*, 2(1):1–10, 2001.

[114] N. R. Jennings and E. H. Mamdani. Using Joint Responsibility to Coordinate Collaborative Problem Solving in Dynamic Environments. In *10th National Conference on Artificial Intelligence*, pages 269–275, San Jose, CA, 1992.

[115] A. Jones and M. Sergot. Deontic Logic in the Representation of Law: Towards a Methodology. *Artificial Intelligence and Law*, 1(1):45–64, 1992.

[116] A. Jones and M. Sergot. On the Characterisation of Law and Computer Systems: The Normative Systems Perspective. In J.-J. Meyer and R.J. Wieringa, editors, *Deontic Logic in Computer Science: Normative System Specification*, pages 275–307. John Wiley & Sons, Chichester, England, 1993.

[117] A. Jones and M. Sergot. A Formal Characterisation of Institutionalised Power. *Logic Journal of the IGPL*, 4(3):427–443, 1996.

[118] L. Kaplow. General Characteristics of Rules. In B. Bouckaert and G. De Geest, editors, *Encyclopedia of Law and Economics*, volume Volume V: The Economics of Crime and Litigation, pages 502–528. Edward Elgar, Cheltenham, UK, 2000.

[119] A. Keller and H. Ludwig. Defining and Monitoring Service Level Agreements for dynamic e-Business. In *16th USENIX Systems Administration Conference (LISA'02)*, pages 189–204, Philadelphia, USA, 2002.

[120] G. E. Kersten and G. Lo. Negotiation support systems and software agents in e-business negotiations, 2001.

[121] G. E. Kersten, S. J. Noronha, and J. Teich. Are All E-Commerce Negotiations Auctions? In *Fourth International Conference on the Design of Cooperative Systems (COOP 2000)*, Sophia Antipolis, France, 2000.

[122] S. Kimbrough and S. Moore. On Automated Message Processing in Electronic Commerce and Work Support Systems: Speech Act Theory and Expressive Felicity. *Transactions on Information Systems*, 15(4):321–367, 1997.

[123] J. E. Kittock. Emergent conventions and the structure of multi-agent systems. In L. Nadel and D. Stein, editors, *Lectures in Complex Systems: the proceedings of the 1993 Complex Systems Summer School*, volume VI, Santa Fe Institute, 1993. Addison Wesley.

[124] M. Klein, P. Faratin, H. Sayama, and Y. Bar-Yam. Negotiating Complex Contracts. In C. Castelfranchi and W. L. Johnson, editors, *First International Joint Conference on Autonomous Agents and Multiagent Systems*, pages 753–757, Bologna, Italy, 2002.

[125] J. Knottenbelt. *Contract Related Agents*. PhD thesis, Imperial College of Science, Technology and Medicine, 2006.

[126] J. Knottenbelt and K. Clark. An Architecture For Contract-based Communicating Agents. In *Second European Workshop on Multi-Agent Systems*, Barcelona, Spain, 2004.

[127] J. Knottenbelt and K. Clark. Contract-Related Agents. In F. Toni and P. Torroni, editors, *Computational Logic in Multi-Agent Systems: 6th International Workshop*, volume 3900 of *Lecture Notes in Computer Science*, pages 226–242. Springer, London, England, 2005.

[128] M. Koetsier, P. Grefen, and J. Vonk. Contracts for Cross-Organizational Workflow Management. In *Electronic Commerce and Web Technologies, First International Conference, EC-Web 2000*, pages 110–121, London, UK, 2000. Springer.

[129] M. J. Kollingbaum and T. J. Norman. Supervised Interaction – Creating a Web of Trust for Contracting Agents in Electronic Environments. In C. Castelfranchi and W.L. Johnson, editors, *First International Joint Conference on Autonomous Agents and Multiagent Systems*, volume 1, pages 272–279, Bologna, Italy, 2002. ACM Press.

[130] R. A. Kowalski and M. Sergot. A logic-based calculus of events. *New Generation Computing*, 4(4):319–340, 1986.

[131] P. Linington, Z. Milosevic, J. Cole, S. Gibson, S. Kulkarni, and S. Neal. A unified behavioural model and a contract language for extended enterprise. *Data & Knowledge Engineering*, 51(1):5–29, 2004.

[132] A. Lomuscio, H. Qu, and M. Solanki. Towards verifying compliance in agent-based Web service compositions. In Padgham, Parkes, Müller, and Parsons, editors, *7th Int. Conf. on Autonomous Agents and Multiagent Systems (AAMAS 2008)*, pages 265–272, Estoril, Portugal, 2008.

[133] H. Lopes Cardoso, A. Malucelli, A. P. Rocha, and E. Oliveira. Institutional Services for Dynamic Virtual Organizations. In L. M. Camarinha-Matos, H. Afsarmanesh, and A. Ortiz, editors, *Collaborative Networks and Their Breeding Environments – 6th IFIP Working Conference on Virtual Enterprises (PRO-VE'05)*, pages 521–528. Springer, Valencia, Spain, 2005.

[134] H. Lopes Cardoso and E. Oliveira. Assisting and Regulating Virtual Enterprise Interoperability through Contracts. In K. Fischer, A. Berre, K. Elms, and J. P. Muller, editors, *AAMAS'2005 Workshop Agent-based Technologies and applications for enterprise interOPerability (ATOP)*, pages 1–12, Utrecht, The Netherlands, 2005.

[135] H. Lopes Cardoso and E. Oliveira. Towards an Institutional Environment using Norms for Contract Performance. In M. Pechoucek, P. Petta, and L. Z. Varga, editors, *Multi-Agent Systems and Applications IV – 4th International Central and Eastern European Conference on Multi-Agent Systems (CEEMAS 2005)*, volume 3690 of *Lecture Notes in Artificial Intelligence*, pages 256–265. Springer, Budapest, Hungary, 2005.

[136] H. Lopes Cardoso and E. Oliveira. Virtual Enterprise Normative Framework within Electronic Institutions. 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 14–32. Springer Verlag, Heidelberg, D-69121, Germany, Toulouse, France, 2005.

[137] H. Lopes Cardoso and E. Oliveira. Institutional Reality and Norms: Specifying and Monitoring Agent Organizations. *International Journal of Cooperative Information Systems*, 16(1):67–95, 2007.

[138] H. Lopes Cardoso and E. Oliveira. A Contract Model for Electronic Institutions. In J.S. Sichman, J. Padget, S. Ossowski, and P. Noriega, editors, *Coordination, Organizations, Institutions, and Norms in Agent Systems III*, LNAI 4870, pages 27–40. Springer, 2008.

[139] H. Lopes Cardoso and E. Oliveira. Electronic Institutions for B2B: Dynamic Normative Environments. *Artificial Intelligence and Law*, 16(1):107–128, 2008.

[140] 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.

[141] 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.

[142] H. Lopes Cardoso and E. Oliveira. Adaptive Deterrence Sanctions in a Normative Framework. In *The 2009 IEEE/WIC/ACM International Conference on Intelligent Agent Technology (IAT 2009)*, pages 36–43, Milan, Italy, 2009. IEEE Computer Society.

[143] H. Lopes Cardoso and E. Oliveira. Flexible Deadlines for Directed Obligations in Agent-based Business Contracts (Extended Abstract). In Decker, Sichman, Sierra, and Castelfranchi, editors, *The 8th International Conference on Autonomous Agents and Multiagent Systems (AAMAS 2009)*, pages 1307–1308, Budapest, Hungary, 2009. IFAAMAS.

[144] H. Lopes Cardoso and E. Oliveira. Monitoring Cooperative Business Contracts in an Institutional Environment. In *11th International Conference on Enterprise Information Systems (ICEIS 2009)*, pages 206–211, Milan, Italy, 2009. INSTICC Press.

[145] H. Lopes Cardoso and E. Oliveira. Risk Tolerance and Social Awareness: Adapting Deterrence Sanctions to Agent Populations. In L. Seabra Lopes, N. Lau, P. Mariano, and L. M. Rocha, editors, *14th Portuguese Conference on Artificial Intelligence (EPIA 2009)*, volume LNAI, pages 560–571, Aveiro, Portugal, 2009. Springer.

[146] H. Lopes Cardoso and E. Oliveira. Directed Deadline Obligations in Agent-Based Business Contracts. In J. Padget, A. Artikis, W. Vasconcelos, K. Stathis, V. Torres da Silva, E. Matson, and A. Polleres, editors, *Coordination, Organizations, Institutions, and Norms in Agent Systems V*, LNAI 6069, pages 225–240. Springer, 2010.

[147] 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.

[148] H. Lopes Cardoso, D. D. Teixeira, and E. Oliveira. Enhancing Interoperability: Ontology-Mapping in an Electronic Institution. In K. Fischer, J. P. Muller, J. Odell, and A. J. Berre, editors, *Agent-based Technologies and Applications for Enterprise Interoperability*, LNBIP, pages 47–62. Springer, 2009.

[149] F. López y López and M. Luck. Modelling Norms for Autonomous Agents. In E. Chavez, J. Favela, M. Mejia, and A. Oliart, editors, *Fourth Mexican International Conference on Computer Science*, pages 238–245. IEEE Computer Society, 2003.

[150] F. López y López, M. Luck, and M. d'Inverno. Constraining Autonomy through Norms. In Castelfranchi C. and Johnson W., editors, *First International Joint Conference on Autonomous Agents and Multiagent Systems*, pages 674–681, Bologna, Italy, 2002. ACM Press.

[151] F. López y López, M. Luck, and M. d'Inverno. A normative framework for agent-based systems. *Computational & Mathematical Organization Theory*, 12(2-3):227–250, 2006.

[152] M. Luck, P. McBurney, O. Shehory, and S. Willmott. *Agent Technology: Computing as Interaction (A Roadmap for Agent Based Computing).* AgentLink, 2005.

[153] M. Luck, S. Munroe, F. López y López, and R. Ashri. Trust and Norms for Interaction. In *Proceedings of the IEEE SMC 2004 International Conference on Systems, Man & Cybernetics*, pages 1944–1949. IEEE, The Hague, The Netherlands, 2004.

[154] A. Malucelli. *Ontology-based Services for Agents Interoperability.* PhD thesis, Universidade do Porto, 2006.

[155] A. Malucelli, D. Palzer, and E. Oliveira. Ontology-based Services to help solving the heterogeneity problem in e-commerce negotiations. *Electronic Commerce Research and Applications*, 5(1):29–43, 2006.

[156] S. E. Masten. Contractual Choice. In B. Bouckaert and G. De Geest, editors, *Encyclopedia of Law and Economics*, volume Volume III: The Regulation of Contracts, pages 25–45. Edward Elgar, Cheltenham, UK, 2000.

[157] P. Meyer. A proposed XML standard for contract documents, 2005.

[158] Z. Milosevic and A. Bond. Electronic Commerce on the Internet: What is Still Missing?, 1995.

[159] Z. Milosevic, A. Josang, T. Dimitrakos, and M. A. Patton. Discretionary Enforcement of Electronic Contracts. In *Sixth International Enterprise Distributed Object Computing Conference (EDOC'02)*, pages 39–50, Lausanne, Switzerland, 2002. IEEE Computer Society.

[160] Z. Milosevic, P. F. Linington, S. Gibson, S. Kulkarni, and J. Cole. Inter-organisational collaborations supported by e-contracts. In W. Lamersdorf, V. Tschammer, and S. Amarger, editors, *Building the E-Service Society: E-Commerce, E-Business, and E-Government*, pages 413–429. IFIP, Kluwer, Toulouse, France, 2004.

[161] N. Minsky and V. Ungureanu. Law-Governed Interaction: a Coordination and Control Mechanism for Heterogeneous Distributed Systems. *ACM Transactions on Software Engineering and Methodology*, 9(3):273–305, 2000.

[162] 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.

[163] C. Molina-Jimenez, S. Shrivastava, J. Crowcroft, and P. Gevros. On the Monitoring of Contractual Service Level Agreements. In *The First IEEE International Workshop on Electronic Contracting (WEC)*, Proceedings - First IEEE International Workshop on Electronic Contracting, WEC 2004, pages 1–8, San Diego, CA, United States, 2004. IEEE Computer Society.

[164] C. Molina-Jimenez, S. Shrivastava, E. Solaiman, and J. Warne. Contract Representation for Run-time Monitoring and Enforcement. In J.-Y. Chung and L.-J. Zhang, editors, *IEEE International Conference on E-Commerce*, pages 103–110. IEEE Computer Society, 2003.

[165] S. Neal, J. Cole, P. F. Linington, Z. Milosevic, S. Gibson, and S. Kulkarni. Identifying requirements for Business Contract language: A monitoring perspective. In *Seventh International Enterprise Distributed Object Computng Conference (EDOC'03)*, pages 50–61, Brisbane, Australia, 2003.

[166] P. Noriega. *Agent Mediated Auctions: The Fishmarket Metaphor.* PhD thesis, Universitat Autònoma de Barcelona, 1997.

[167] D. C. North. *Institutions, Institutional Change and Economic Performance.* Cambridge University Press, 1990.

[168] D. Nute. Defeasible Logic. In D.M. Gabbay, C.J. Hogger, and J.A. Robinson, editors, *Handbook of Logic in Artificial Intelligence and Logic Programming*, volume 3, pages 353–395. Oxford University Press, 1994.

[169] D. Nute. *Defeasible Deontic Logic*, volume 263 of *Synthese Library*. Kluwer Academic Publishers, 1997.

[170] J. Odell, H. V. D. Parunak, M. Fleischer, and S. Breuckner. Modeling Agents and Their Environment. In F. Giunchiglia, J. Odell, and G. Weiss, editors, *Agent-Oriented Software Engineering III*, volume 2585 of *Lecture Notes in Computer Science*, pages 16–31. Springer, 2002.

[171] A. I. Oliveira, L. M. Camarinha-Matos, and M. Pouly. Agreement Negotiation Support in VO Creation. In L. Camarinha-Matos and W. Picard, editors, *Pervasive Collaborative Networks*, pages 107–118. Springer, 2008.

[172] E. Oliveira and A. P. Rocha. Agents Advanced Features for Negotiation in Electronic Commerce and Virtual Organisations Formation Process. In F. Dignum and C. Sierra, editors, *Agent Mediated Electronic Commerce: The European AgentLink Perspective*, pages 78–97. Springer, 2000.

[173] N. Oren, S. Panagiotidi, J. Vázquez-Salceda, S. Modgil, M. Luck, and S. Miles. Towards a Formalisation of Electronic Contracting Environments. In *Coordination, Organizations, Institutions and Norms in Agent Systems IV*, Lecture Notes in Computer Science, pages 156–171. Springer, 2009.

[174] S. Panagiotidi, J. Vázquez-Salceda, S. Álvarez Napagao, S. Ortega-Martorell, S. Willmott, R. Confalonieri, and P. Storms. Intelligent Contracting Agents Language. In *AISB 2008 Convention – Communication, Interaction and Social Intelligence*, volume Volume 4: Proceedings of the AISB 2008 Symposium on Behaviour Regulation in Multi-agent Systems, pages 49–54, Aberdeen, UK, 2008.

[175] 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.

[176] S. A. Petersen. *An Agent-based Approach to Support the Formation of Virtual Enterprises*. PhD thesis, Norwegian University of Science and Technology, 2003.

[177] A. M. Polinsky and S. Shavell. Punitive Damages. In P. Newman, editor, *The New Palgrave Dictionary of Economics and the Law*, volume 3, pages 192–198. Stockton Press, New York, 1998.

[178] C. Preist. Agent Mediated Electronic Commerce Research at Hewlett Packard Labs. *SIGecom Exchanges*, 2.3:18–28, 2001.

[179] G. Quirchmayr, Z. Milosevic, R. Tagg, J. Cole, and S. Kulkarni. Establishment of Virtual Enterprise Contracts. In R. Cicchetti, A. Hameurlain, and R. Traunmüller, editors, *Database and Expert Systems Applications: 13th International Conference (DEXA 2002)*, pages 236–248. Springer, 2002.

[180] C. Quix, M. Schoop, and M. Jeusfeld. Business Data Management for Business-to-Business Electronic Commerce. *ACM SIGMOD Record*, 31(1):49–54, 2002.

[181] H. Raiffa. *The Art and Science of Negotiation*. Harvard University Press, 1982.

[182] A. S. Rao and M. P. Georgeff. Modeling Rational Agents within a BDI-Architecture. In J. Allen, R. Fikes, and E. Sandewall, editors, *Second International Conference on Principles of Knowledge Representation and Reasoning (KR-91)*, pages 473–484. Morgan Kaufmann, 1991.

[183] D. M. Reeves, M. P. Wellman, and B. N. Grosof. Automated Negotiation from Declarative Contract Descriptions. *Computational Intelligence*, 18:482–500, 2002.

[184] R. Reiter. A logic for default reasoning. *Artificial Intelligence*, 13(1/2):81–132, 1980.

[185] A. P. Rocha. *Metodologias de Negociação em Sistemas Multi-Agentes para Empresas Virtuais*. PhD thesis, Universidade do Porto, 2001.

[186] A. P. Rocha, H. Lopes Cardoso, and E. Oliveira. Contributions to an Electronic Institution supporting Virtual Enterprises' Life cycle. In G. D. Putnik and M. M. Cunha, editors, *Virtual Enterprise Integration: Technological and Organizational Perspectives*, pages 229–246. Idea Group Inc., 2005.

[187] A. P. Rocha and E. Oliveira. Electronic Institutions as a framework for Agents' Negotiation and mutual Commitment. In P. Brazdil and A. Jorge, editors, *Progress in Artificial Intelligence (Proceedings of 10th EPIA)*, volume 2258 of *LNAI*, pages 232–245, Porto, Portugal, 2001. Springer.

[188] J. A. Rodríguez-Aguilar. *On the design and construction of Agent-mediated Electronic Institutions*. PhD thesis, Universitat Autònoma de Barcelona, 2001.

[189] D. Romero, J. Giraldo, N. Galeano, and A. Molina. Towards Governance Rules and Bylaws for Virtual Breeding Environments. In L.M. Camarinha-Matos, H. Afsarmanesh, P. Novais, and C. Analide, editors, *Establishing the Foundation of Collaborative Networks*, International Federation for Information Processing (IFIP), pages 93–102. Springer, 2007.

[190] D. Romero, A. I. Oliveira, L. M. Camarinha-Matos, and A. Molina. The Virtual Enterprise from a Governance Perspective. In L.M. Camarinha-Matos, P. Pereira, and L. Ribeiro, editors, *Emerging Trends in Technological Innovation*, pages 73–82. Springer, 2010.

[191] J. S. Rosenschein and G. Zlotkin. *Rules of Encounter: Designing Conventions for Automated Negotiation among Computers.* MIT Press, Cambridge, MA, 1994.

[192] L. Royakkers and F. Dignum. Defeasible reasoning with legal rules. In M.A. Brown and J. Carmo, editors, *Deontic Logic, Agency and Normative Systems (Workshops in Computing)*, pages 174–193. Springer-Verlag, 1996.

[193] A. Runge, B. Schopp, and K. Stanoevska-Slabeva. The Management of Business Transactions through Electronic Contracts. In *10th International Workshop on Database and Expert Systems Applications (DEXA'99)*, pages 824–831, Florence, Italy, 1999. IEEE Computer Society.

[194] Y. U. Ryu. Relativized Deontic Modalities for Contractual Obligations in Formal Business Communication. In *30th Hawaii International Conference on System Sciences (HICSS)*, volume 4, pages 485–493, Hawaii, USA, 1997.

[195] N. J. Saam and A. Harrer. Simulating Norms, Social Inequality, and Functional Change in Artificial Societies. *Journal of Artificial Societies and Social Simulation*, 2(1), 1999.

[196] M. Sallé. Electronic Contract Framework for Contractual Agents. In R. Cohen and B. Spencer, editors, *Advances in Artificial Intelligence: 15th Conference of the Canadian Society for Computational Studies of Intelligence*, pages 349–353. Springer, 2002.

[197] F. Santos and J. Carmo. A Deontic Logic Representation of Contractual Obligations. In J.-J. Meyer and R.J. Wieringa, editors, *Deontic Logic in Computer Science: Normative System Specification*, pages 243–257. John Wiley & Sons, Chichester, England, 1993.

[198] G. Sartor. Normative conflicts in legal reasoning. *Artificial Intelligence and Law*, 1(2-3):209–235, 1992.

[199] M. Schoop and C. Quix. DOC.COM: a framework for effective negotiation support in electronic marketplaces. *Computer Networks*, 37(2):153–170, 2001.

[200] C. V. Schoubroeck, H. Cousy, B. Windey, and D. Droshout. A Legal Taxonomy on Virtual Enterprises. In K.-D. Thoben, F. Weber, and K. S. Pawar, editors, *7th International Conference on Concurrent Enterprising*, pages 357–364, Bremen, Germany, 2001.

[201] J. Scott. *Internalization of Norms: A Sociological Theory of Moral Commitment*. Prentice-Hall, Englewoods Cliffs, N.J., 1971.

[202] J. R. Searle. *Speech Acts: an Essay in the Philosophy of Language*. Cambridge University Press, Cambridge, 1969.

[203] J. R. Searle. *The Construction of Social Reality*. Free Press, New York, 1995.

[204] S. Sen and S. Airiau. Emergence of norms through social learning. In *20th International Joint Conference on Artificial Intelligence*, pages 1507–1512, Hyderabad, India, 2007.

[205] M. Sergot. A Computational Theory of Normative Positions. *ACM Transactions on Computational Logic*, 2(4):581–622, 2001.

[206] M. J. Sergot. The representation of law in computer programs: A survey and comparison. In T.J.M. Bench-Capon, editor, *Knowledge Based Systems and Legal Applications*. Academic Press, London, 1990.

[207] M. Shanahan. The event calculus explained. In M.J. Wooldridge and M. Veloso, editors, *Artificial Intelligence Today*, Lecture Notes in Artificial Intelligence, pages 409–430. Springer Verlag, 1999.

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

[209] Y. Shoham and M. Tennenholtz. On the emergence of social conventions: modeling, analysis, and simulations. *Artificial Intelligence*, 94(1-2):139–166, 1997.

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

[211] J. M. Swaminathan, S. F. Smith, and N. M. Sadeh. Modeling Supply Chain Dynamics: A Multiagent Approach. *Decision Sciences*, 29(3):607–632, 1998.

[212] Y.-H. Tan and W. Thoen. Modeling Directed Obligations and Permissions in Trade Contracts. In *Proceedings of the Thirty-First Annual Hawaii International Conference on System Sciences, Volume 5*. IEEE Computer Society, 1998. 798835 166.

[213] Y.-H. Tan and W. Thoen. DocLog: An Electronic Contract Representation Language. In *11th International Workshop on Database and Expert Systems Applications (DEXA'00)*, pages 1069–1073, London, U.K., 2000.

[214] The European Parliament and the Council of the European Union. Directive 2000/31/EC on Electronic Commerce. *Official Journal of the European Communities*, 43(L 178):1–16, 2000.

[215] N. A. M. Tinnemeier, M. M. Dastani, J.-J. Ch. Meyer, and L. van der Torre. Programming Normative Artifacts with Declarative Obligations and Prohibitions. In *2009 IEEE/WIC/ACM International Joint Conference on Web Intelligence and Intelligent Agent Technology*, pages 145–152, Milan, Italy, 2009. IEEE Computer Society.

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

[217] United Nations Commission on International Trade Law. United Nations Convention on Contracts for the International Sale of Goods (CISG), 1980.

[218] J. Urbano, H. Lopes Cardoso, and E. Oliveira. Making Electronic Contracting Operational and Trustworthy. In *12th Ibero-American Conference on Artificial Intelligence*, Bahía Blanca, Argentina, 2010. Springer.

[219] J. Urbano, A. P. Rocha, and E. Oliveira. Computing Confidence Values: Does Trust Dynamics Matter? In L. Seabra Lopes, N. Lau, P. Mariano, and L. M. Rocha, editors, *14th Portuguese Conference on Artificial Intelligence (EPIA 2009)*, volume LNAI, pages 520–531, Aveiro, Portugal, 2009. Springer.

[220] P. Valckenaers, J. Sauter, C. Sierra, and J. A. Rodríguez-Aguilar. Applications and environments for multi-agent systems. *Autonomous Agents and Multi-Agent Systems*, 14(1):61–85, 2007.

[221] L. van der Torre. Violated Obligations in a Defeasible Deontic Logic. In *Proceedings of the Eleventh European Conference on Artificial Intelligence (ECAI'94)*, pages 371–375, Amsterdam, The Netherlands, 1994. John Wiley & Sons.

[222] B. C. van Fraassen. The logic of conditional obligation. *Journal of Philosophical Logic*, 1:417–438, 1972.

[223] M. Verdicchio and M. Colombetti. Commitments for Agent-Based Supply Chain Management. *ACM SIGecom Exchanges*, 3(1):13–23, 2002.

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

[225] G. von Wright. Deontic logic. *Mind*, 60:1–15, 1951.

[226] J. Vázquez-Salceda. *The role of Norms and Electronic Institutions in Multi-Agent Systems applied to complex domains. The HARMONIA framework.* PhD thesis, Universitat Politècnica de Catalunya, 2003.

[227] 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.

[228] J. Vázquez-Salceda and F. Dignum. Modelling Electronic Organizations. In V. Marik, J. Muller, and M. Pechoucek, editors, *Multi-Agent Systems and Applications III: 3rd. International Central and Eastern European Conference on Multi-Agent Systems (CEEMAS'03)*, volume 2691 of *Lecture Notes in Artificial Intelligence*, pages 584–593, Prague, Czech Republic, 2003. Springer Verlag.

[229] A. Walker and M. Wooldridge. Understanding the Emergence of Conventions in Multi-Agent Systems. In V. Lesser and L. Gasser, editors, *Proceedings of the First International Conference on Multi-Agent Systems*, pages 384–389. MIT Press, San Francisco, USA, 1995.

[230] F. Wan and M. P. Singh. Formalizing and achieving multiparty agreements via commitments. In *Proceedings of the fourth international joint conference on Autonomous agents and multiagent systems*, pages 770–777, The Netherlands, 2005. ACM. 1082591 770-777.

[231] H. Weigand, M. Schoop, A. de Moor, and F. Dignum. B2B Negotiation Support: the need for a communication perspective. *Group Decision and Negotiation*, 12(1):3–29, 2003.

[232] D. Weyns and T. Holvoet. A Formal Model for Situated Multi-Agent Systems. *Fundamenta Informaticae*, 63(2-3):125–158, 2004.

[233] D. Weyns, A. Omicini, and J. Odell. Environment as a first class abstraction in multiagent systems. *Journal of Autonomous Agents and Multi-Agent Systems*, 14(1):5–30, 2007.

[234] D. Weyns, H. V. D. Parunak, F. Michel, T. Holvoet, and J. Ferber. Environments for Multiagent Systems, State-of-the-Art and Research Challenges. In D. Weyns, H. V. D. Parunak, and F. Michel, editors, *Environments for Multi-Agent Systems*, volume 3374 of *Lecture Notes in Computer Science*, pages 1–47. Springer, 2005.

[235] R. Wieringa and J.-J. Meyer. Applications of deontic logic in computer science: a concise overview. In J.-J. Meyer and R.J. Wieringa, editors, *Deontic Logic in Computer Science: Normative System Specification*, pages 17–40. John Wiley & Sons, Chichester, England, 1993.

[236] M. Wooldridge and N. R. Jennings. Intelligent Agents: Theory and Practice. *Knowledge Engineering Review*, 10(2):115–152, 1995.

[237] P. R. Wurman, M. P. Wellman, and W. E. Walsh. The Michigan Internet AuctionBot: A configurable auction server for human and software agents. In K. P. Sycara and M. Wooldridge, editors, *Second International Conference on Autonomous Agents*, pages 301–308, Minneapolis, MN, USA, 1998. ACM, New York, NY, USA.

[238] L. Xu. Monitorable Electronic Contract. In *The 2003 IEEE International Conference on E-Commerce (CEC'03)*, pages 92–99. IEEE Computer Society Press, 2003.

[239] L. Xu. *Monitoring Multi-party Contracts for E-business*. PhD thesis, University of Tilburg, 2004.

[240] H. Yoshino. Logical Structure of Contract Law System - For Constructing a Knowledge Base of the United Nations Convention on Contracts for the International Sale of Goods -. *Journal of Advanced Computational Intelligence*, 2(1):2–11, 1998.

[241] F. C. Zagare and D. M. Kilgour. *Perfect Deterrence*. Cambridge University Press, 2000.