THESIS

PROGRAMA DE DOCTORADO DE RECONOCIMIENTO DE FORMAS E INTELIGENCIA ARTIFICIAL

Using Norms To Control Open Multi-Agent Systems

Dissertation submitted by: Natalia Criado Pacheco

Supervisors: Dra. Estefanía Argente Villaplana,

Dr. Vicente Botti Navarro and

Dr. Pablo Noriega

Grupo de Tecnología Informática - Inteligencia Artificial

Departamento de Sistemas Informáticos y Computación

Universidad Politécnica de Valencia

Camino de Vera, s/n

46020 Valencia, Spain

Contents

List of Figures

List of Tables

Abstract

Internet is, maybe, the most relevant scientific advance of our days. It has also allowed the evolution of traditional computational paradigms into the paradigm of distributed computation over an open network of machines. Multi-agent systems (MAS) have been proposed as a suitable technology for addressing challenges motivated by these open distributed systems. MAS applications are formed by agents which may be designed independently according to different goals and motivations. Therefore, no assumption about their behaviours can be made *a priori*. Because of this, coordination and cooperation mechanisms, such as *norms*, are needed in MAS for ensuring social order and avoiding conflicts.

The notion of *norm* covers mainly two different dimensions: i) norms as an *instrument* for guiding citizens when performing actions and activities, so norms define which procedures, or protocols must be followed in a concrete situation; and ii) norms as *orders* or *prohibitions* supported by threats of sanction, thus norms are means to prevent or punish certain actions. In MAS research, norms have been defined as a formal specification of what is permitted, obliged and forbidden within a society. Thus, they aim at regulating the life of software agents and the interactions among them.

The main objective of this thesis is to allow MAS designers to use norms as a mechanism for controlling and coordinating open MAS. We aim to develop norm-based mechanisms for MAS at two levels: agent models and agent infrastructures. Thus, in this thesis we first address the problem of defining norm-autonomous agents that deliberate about norms within uncertain environments. Secondly, in this thesis we propose a distributed architecture for enforcing norms in open MAS, named MaNEA, which has been integrated into the Magentix2 platform. This proposed architecture implements norms in an optimized way, given that in open MAS the internal states of agents are not accessible. Therefore, norms cannot be imposed as agent's beliefs or goals, but they must be implemented in the platform by means of control mechanisms.

Resumen

Internet es, tal vez, el avance científico más relevante de nuestros días. Entre otras cosas, Internet ha permitido la evolución de los paradigmas de computación tradicionales hacia el paradigma de computación distribuida, que se caracteriza por utilizar una red abierta de ordenadores. Los sistemas multi-agente (SMA) son una tecnología adecuada para abordar los retos motivados por estos sistemas abiertos distribuidos. Los SMA son aplicaciones formadas por agentes heterogéneos y autónomos que pueden haber sido diseñados de forma independiente de acuerdo con objetivos y motivaciones diferentes. Por lo tanto, no es posible realizar ninguna hipótesis *a priori* sobre el comportamiento de los agentes. Por este motivo, los SMA necesitan de mecanismos de coordinación y cooperación, como las *normas*, para garantizar el orden social y evitar la aparición de conflictos.

El término *norma* cubre dos dimensiones diferentes: i) las normas como un *instrumento* que guía a los ciudadanos a la hora de realizar acciones y actividades, por lo que las normas definen los procedimientos y/o los protocolos que se deben seguir en una situación concreta, y ii) las normas como *órdenes* o *prohibiciones* respaldadas por un sistema de sanciones, por lo que las normas son medios para prevenir o castigar ciertas acciones. En el área de los SMA, las normas se vienen utilizando como una especificación formal de lo que está permitido, obligado y prohibido dentro de una sociedad. De este modo, las normas permiten regular la vida de los agentes software y las interacciones entre ellos.

La motivación principal de esta tesis es permitir a los diseñadores de los SMA utilizar normas como un mecanismo para controlar y coordinar SMA abiertos. Nuestro objetivo es elaborar mecanismos normativos a dos niveles: a nivel de agente y a nivel de infraestructura. Por lo tanto, en esta tesis se aborda primero el problema de la definición de agentes normativos autónomos que sean capaces de deliberar acerca de las normas dentro de entornos inciertos. En segundo lugar, en esta tesis se propone una arquitectura distribuida, llamada MaNEA, que permite la monitorización e implementación de las normas en SMA abiertos. Dicha arquitectura se ha

integrado en la plataforma de agentes Magentix2. Dado que en los SMA los estados internos de los agentes no son accesibles, las normas no se pueden imponer como creencias u objetivos y deben ser implementadas por las plataformas de agentes mediante mecanismos de control.

Resum

Internet és, potser, l'avanç científic més rellevant dels nostres dies. Entre altres coses, Internet ha permès l'evolució dels paradigmes de computació tradicionals cap al paradigma de computació distribuïda, que es caracteritza per utilitzar una xarxa oberta d'ordinadors. Els sistemes multi-agent (SMA) són una tecnologia adequada per abordar els reptes motivats per aquests sistemes oberts distribuïts. Els SMA són aplicacions formades per agents heterogenis i autònoms que poden haver estat dissenyats de forma independent d'acord amb objectius i motivacions diferents. Per tant, no es pot fer cap hipòtesi *a priori* sobre el comportament dels agents. Per aquest motiu, els SMA necessiten de mecanismes de coordinació i cooperació, com les *normes*, per garantir l'ordre social i evitar conflictes.

El terme *norma* cobreix dues dimensions diferents: i) les normes com un *instrument* que guia els ciutadans a l'hora de realitzar accions i activitats, de manera que les normes defineixen els procediments i/o els protocols que s'han de seguir en una situació concreta, i ii) les normes com *ordres* o *prohibicions* recolzades per un sistema de sancions, de manera que les normes són mitjans per prevenir o castigar certes accions. En l'àrea dels SMA, les normes s'han utilitzat com una especificació formal del que està permès, obligat i prohibit dins d'una societat. D'aquesta manera, les normes permeten regular la vida dels agents sofware i les interaccions entre ells.

La motivació principal d'aquesta tesi és permetre als dissenyadors dels SMA utilitzar normes com un mecanisme per controlar i coordinar SMA. El nostre objectiu és elaborar mecanismes normatius a dos nivells: a nivell d'agent i a nivell d'infraestructura. Per tant, en aquesta tesi s'aborda el problema de la definició d'agents normatius autònoms que siguin capaços de deliberar sobre les normes dins d'entorns incerts. D'altra banda, en aquesta tesi es proposa una arquitectura distribuïda, anomenada MaNEA, que permet la monitorització i implementació de les normes en SMA oberts. Aquesta arquitectura s'ha integrat en la plataforma d'agents Magentix2. Atès que en els SMA els estats interns dels agents no són accessibles, les normes no es poden imposar com creences o objectius i han de ser implementades per les plataformes

d'agents mitjançant mecanismes de control.

Chapter 1

Introduction

Internet is, maybe, the most relevant scientific advance of our days. It has deeply impacted the way in which humans work, entertain themselves and learn. Internet has also allowed the evolution of traditional computational paradigms, in which problems are solved by an isolated machine; into the paradigm of distributed computation over a network of machines. This new paradigm, known as "Computing as Interaction" [LMSW05] proposes the solution of problems by means of the communication among heterogeneous software entities. Artificial Intelligence (AI) in general and multi-agent systems (MAS) in particular have been proposed as a suitable technology for addressing those challenges motivated by these complex and dynamic systems. A Multi-agent System (MAS) consists of a number of agents that interact with one-another [Woo02]. According to [WJ95], an agent is defined by its flexibility, which implies that an agent is: reactive, an agent must answer to its environment; proactive, an agent has to be able to try to fulfil his own plans or objectives; and social, an agent has to be able to communicate with other agents by means of some kind of language. Open MAS are characterized by the heterogeneity of their participants, non-trustworthy members, existence of conflicting individual goals and a high possibility of non-accordance with specifications [AP01]. The main feature of agents in open MAS is autonomy. It is this autonomy that requires coordination and cooperation mechanisms for ensuring social order and avoiding conflicts. With this aim, "social" notions, such as *norms*, have been introduced in the MAS research.

Norms are a coordination mechanism that attempts to promote behaviours that are satisfactory to the organization, i.e., actions that contribute to the achievement of global goals; and avoid harmful actions, i.e., actions that prompt the system to be unsatisfactory or unstable.

The *norm* concept is defined by the Encyclopaedia Britannica [1]as:

> "a rule or standard of behaviour shared by members of a social group. Norms may be internalized; i.e., incorporated within the individual so that there is conformity without external rewards or punishments, or they may be enforced by positive or negative sanctions from without. [...] Norms are more specific than values or ideals: honesty is a general value, but the rules defining what is honest behaviour in a particular situation are norms"

According to this definition, norms guide the behaviour of the members of a group; i.e., they are aimed at achieving coordination inside this group. The notion of norm covers two different dimensions: i) norms as an *instrument* for guiding citizens when performing actions and activities, so norms define which procedures and protocols must be followed in a concrete situation; and ii) norms as *orders* or *prohibitions* supported by threats of sanction, thus norms are means to prevent or punish certain actions. In the MAS research scene, norms have been defined as a formal specification aimed at controlling and coordinating the life of software agents and the interactions among them [RS08]. Norms prescribe what is permitted, forbidden, and mandatory in agent societies. Thus, they define the benefits and the responsibilities of the society's members and, as a consequence, agents are able to plan their actions according to the behaviour expected from the other members. However, norms are not only regulations, but they also establish social institutions which give rise to new types of facts [Sea69]. In general, processes that require coordination and cooperation also require the definition of norms that control the interactions [LyLL02]. Normative multi-agent systems (NMAS) are MAS that use norms as a mechanism for persuading autonomous and heterogeneous agents to behave according to the stated social order [BvdTV08a]. Therefore, NMAS define norms, which are immaterial entities that exist thanks to their acceptance by the society members, in order to avoid conflicts and ensure social order [BvdTV07].

1.1 Motivation

In spite of the great amount of work on norms in MAS, there are many issues that are still pending. Many of these issues are due to the specific challenges of open MAS. As previously

[1]norm. (2010). In Encyclopaedia Britannica. Retrieved November 17, 2010, from Encyclopaedia Britannica Online: http://www.britannica.com/EBchecked/topic/418203/norm

mentioned, open MAS are composed of heterogeneous agents that interact with each other to solve a complex problem in a distributed way. The characteristic features of the environment in which agent interactions take place are key factors for the creation of norms that control these interactions. In this sense, one of the most relevant properties of the environments where agents interact is the *uncertainty*. There may be different reasons for uncertainty. In this thesis we will focus on the following: (i) the environment in which agents interact may change drastically, this implies that norms may need to be dynamically adapted in response to these changes; (ii) agents may be designed independently (even by different parties) according to different goals and motivations and no assumption about agent behaviours can be made *a priori*; (iii) agents have a limited and not fully believable knowledge of the world; (iv) there may be ambiguous interpretations of the norms causing doubts, conflicts or confusion. Uncertainty (that usually characterize open MAS) has received little attention in the existing literature on norms and MAS. However, it is a very relevant issue that must be considered inside the MAS field. In this thesis, we focus on these issues when using norms to control MAS. In this sense, this work is aimed at developing both an agent architecture that takes into account norms and a norm-enforcing system that takes into account the features of open MAS.

There are few works focused on normative reasoning from an individual perspective. The usage of norms as formal statements aimed at regulating agent societies entails the development of intelligent norm-autonomous agents. What is the point of defining norms for controlling MAS if there is no agent capable of considering them? The norm acceptance problem [CCD99] consists of two main problems: the recognition of norms as such inside agents' minds; and the norm compliance decision, i.e., the consideration of these norms in agents' decision making process. The set of norms that regulate a particular MAS may dynamically evolve along time. Therefore, agents must be able to recognise and adopt new norms but maintaining their autonomy. Existing proposals of intelligent norm-aware agents, like [KN03, SST06, BDH$^+$01], tend to be concerned about the decision-making processes that are supported by a set of active norms whose validity is taken for granted. Thus, they consider norms as static constraints that are hard-wired on agents. While a few, like [ACCC08], treat methods that allow agents to recognise the set of norms that control their environment, only a fraction have been concerned about the two issues above plus the uncertainty of the environment. In this thesis, we address the problem of defining norm-aware agents and, in particular, we discuss how these agents deliberate about norms within uncertain environments.

The existence of norm-autonomous agents that are capable of violating norms entails the development of norm-enforcing system that implement norms, given that in open MAS the internal states of agents are not accessible [CAB11a]. Therefore, norms cannot be imposed as agent's beliefs or goals, but they must be implemented by means of control mechanisms. In this thesis we propose an norm-enforcing system that has been designed to overcome the main drawbacks that the existing agent platforms and infrastructures present when they are used to control norms in open MAS.

This thesis has been developed under the frame of three research projects on Multi-agent Systems. The development of use of norms for controlling MAS is a common and transversal topic in all of these projects. Moreover, we incorporated some of the main results of this thesis directly in the models and infrastructures developed in these projects. Specifically, this thesis is developed under the frame of the following projects funded by the Spanish Government:

- "Thomas: MeTHods, Techniques and Tools for Open Multi-Agent Systems" under grant TIN2006-14630-C03-01 (Main Researcher: Vicente Botti Navarro, from 2006 to 2009). The main goal of this project is the development of techniques and methods suitable for the creation of open MAS that are capable of solving problems in an autonomous and flexible way. These systems are characterized by the heterogeneity of their participants; their limited trust; a high uncertainty; and the existence of individual goals that might be in conflict. In these scenarios, norms are conceived as an effective mechanism for achieving coordination and ensuring social order.

- "Magentix2: A Multiagent Platform for Open Multiagent Systems" under grant TIN2008-04446 (Main Researcher: Ana Garcia-Fornes, from 2008 to 2011). Magentix2 is an agent platform that supports the development and execution of open MAS. Norms must also be considered in the design and implementation of agent platforms. Thus, we extended Magentix2 to implement norms in an optimized way, given that in open MAS the internal states of agents are not accessible.

- "Agreement Technologies" CONSOLIDER-INGENIO 2010 under grant CSD2007-00022 (Main Researcher: Carles Sierra, from 2007 to 2012). Agreement Technologies (AT) refer to computer systems in which autonomous software agents negotiate with one another, typically on behalf of humans, in order to come to mutually acceptable agreements. Norms have been widely promoted as an approach to coordinate multi-agent interactions. This

entails the development of a model of agent capable of taking decisions autonomously, equipped with complex decision-making mechanisms that allow agents to reason about norm adoption and compliance.

1.2 Objectives

The main objective of this thesis is to allow MAS designers to use norms as a mechanism for controlling and coordinating open MAS. We aim to develop norm-based mecanisms for open MAS at two levels: agent models and agent infrastructures. To fulfil this general objective, we deal with the following sub-objectives:

O.1 To survey, classify, and review the existing literature on norms and MAS, and to identify open challenges in this field.

O.2 To propose and validate an agent architecture that allows agents to reason about norms. This general objective entails the following subobjectives:

O.2.1 To propose and validate an agent architecture that allows agents to represent norms and instances explicitly.

O.2.2 To propose and validate mechanisms for allowing agents to reason about norm acceptance and relevance.

O.2.3 To propose and validate mechanisms for allowing agents to reason about compliance with deontic norms.

O.2.4 To propose and validate mechanisms for allowing agents to reason about constitutive norms.

O.2.5 To propose and validate mechanisms for allowing agents to resolve conflicts among norms and other mental propositions.

O.3 To propose and validate a norm-enforcing system that overcomes the main deficiencies and drawbacks of agent platforms and infrastructures when supporting norms in open MAS.

O.4 To integrate our proposed norm-enforcing system and agent architecture into an agent platform.

1.3 Contributions

The specific contributions of this thesis are:

- **State of the Art**. To achieve the first objective **O.1**, in this thesis we review the most relevant works on norms for MAS. This review considers open MAS challenges and points out the main open questions that remain in norm representation, reasoning, creation, and implementation.

- **Norm-Autonomous Agent Architecture**. Regarding the challenge of building norm-aware agents (formulated in objectives **O.2.1** and **O.2.2**), in this thesis we extend the graded multi-context BDI agent architecture [CGS11] with an acquisition context and a compliance context to allow agents to acquire norms from their environment and determine when they are relevant.

- **Reasoning Techniques for Deontic Norms**. Besides the explicit representation of norms, we also contribute techniques for agents to deliberate about the convenience of norm obedience (objective **O.2.3**). Deliberating about norm compliance not only implies considering reasons for and against norm fulfilment but also for and against norm violation. The deliberated and rational violation of norms is a conduct which can be observed in all human societies. Moreover, Castelfranchi [Cas03] claimed that there is not any organization which has been successful without a coordinated and systematic violation on norms. Finally, the role of emotions in the norm compliance dilemma has been analysed and validated.

- **Reasoning Techniques for Constitutive Norms**. Agents may become members of different institutions along its life. Thus, agents need capabilities that allow them to determine the repercussion that their actions may have in different institutions. This anchorage between the real world and the institutional world is defined by means of constitutive norms. This thesis also considers the role of constitutive norms in agent reasoning (objective **O.2.4**).

- **Coherence-Based Mechanism for Solving Conflicts**. Agents may be affected by norms that are in conflict with their cognitive elements. They may even be affected by conflicting norms. Hence, agents should resolve contradictions before making a decision

about which action to perform (objective **O.2.5**). In this thesis, we propose a coherence-based mechanism that solves the existence of conflicting propositions by calculating and selecting those propositions that maximize the coherence of the cognition set.

- **Norm-Enforcing System**. To fulfil objective **O.3**, we propose a new Norm-Enforcing system aimed at controlling open MAS. Specifically, we integrated this system into the Magentix2 platform (objective **O.4**). This system monitors and enforces norms, since in open MAS the internal states of agents are not accessible and norms cannot be imposed as agent's beliefs or goals.

- **Implementation of the Norm-Autonomous Agent Architecture**. Finally, we describe the prototype of the agent architecture that we developed using Jason and Magentix2 (objective **O.4**).

1.4 Document Structure

This document is structured in two main parts. The first part, which consists of Chapters 2 and 3 presents backgrounds of this thesis. Specifically, Chapter 2 provides an overview of the state of the art on the definition of norms for controlling agent societies. Moreover, some basic definitions used in this thesis are provided in Chapter 3.

In the second part of this document the thesis proposal is presented. Chapter 4 describes the norm-autonomous agent architecture developed in this thesis. Chapter 5 describes how agents reason about deontic norms. Chapter 6 focuses on mechanisms for reasoning about constitutive norms. In Chapter 7, we propose a mechanism for resolving conflicts among norms and other mental propositions. In Chapter 8, we detail a case study of our agent architecture. Chapter 9 introduces the norm-enforcing system and the implementation of a prototype of the agent architecture. Finally, we present our concluding remarks in Chapter 10.

Chapter 2

State of the Art

In general, norms represent an effective tool for achieving coordination and cooperation among the members of a society. They have been employed in the field of Artificial Intelligence as a formal specification of deontic statements aimed at regulating the actions of software agents and the interactions among them. A challenging problem currently addressed in the multi-agent systems area is the development of open systems; which are characterized by the heterogeneity and the dynamic features of both their participants and their environment. The main feature of agents in these systems is autonomy. It is this autonomy that requires regulation, and norms are a solution for this. This chapter gives an overview of the most relevant works on norms for multi-agent systems. This review considers open multi-agent systems challenges and points out the main open questions that remain in norm representation, reasoning, creation, and implementation.

2.1 Introduction

The *norm* concept is an ambiguous term that has been given different meanings. In a general sense, norms have been defined as a mechanism for organizing and controlling a society [Pos96]. According to this view of norms, computer systems have been abstracted as systems of norms (i.e., *normative systems*):

> "law, computer systems, and many other kinds of organisational structure may be viewed as instances of normative systems ... Norms prescribe how the agents ought to behave, and specify how they are permitted to behave and what their rights are." [JS93]

Norms have been employed in Artificial Intelligence (AI) research as a formal specification of deontic statements that aim at regulating the life of software entities and the interactions among them. Specifically, norms have been proposed in the AI field to deal with coordination issues and security issues in multi-agent systems (MAS), as well as to model legal issues in electronic institutions and electronic commerce, among other issues.

The most promising application of MAS technology is its use for supporting open distributed systems [LM08]. Open systems are characterized by the heterogeneity of their participants, non-trusty members, existence of conflicting individual goals and a high possibility of nonaccordance with specifications [AP01]. The main feature of agents in these systems is autonomy. It is this autonomy that requires regulation, and norms are a solution for this requirement. In these types of systems, problems are solved by means of cooperation among several software agents [LMSW05]. Norms prescribe what is permitted, forbidden, and mandatory in societies. Thus, they define the benefits and responsibilities of the society members and, as a consequence, agents are able to plan their actions according to their expected behaviour. In general, any process that requires coordination and cooperation also requires the definition of norms that control this interaction [LyLLd02]. Therefore, *normative multi-agent systems* (NMAS) have been defined as MAS that use norms as a mechanism for persuading autonomous and heterogeneous agents to behave according to the stated social order [BvdTV08a]. Therefore, NMAS define norms, which are immaterial entities that exist thanks to their acceptance by the society members, in order to avoid conflicts and ensure social order [BvdTV07].

In spite of the great amount of work that has been done using norms in MAS, there are many issues related to the complexity of open systems that are still pending. This chapter gives an overview of the most relevant works on norms for MAS. This review points out the main deficiencies and drawbacks of current proposals with reference to the specific challenges of open systems. This chapter is structured as follows: Section 2.2 gives a brief introduction to works on norms from a sociological, philosophical, and AI & Law perspective. Sections 2.3 to 2.7 are focused on the main issues in the use of norms for MAS, which are the definition, representation, reasoning, implementation and creation of norms, respectively. The issues and proposals described in each section are connected. Thus, there are open issues that may belong to more than one section. For this reason, Section 2.8 contains a summary of open issues for NMAS.

2.2 Norm Approaches

The role of *norms* in human societies has been analysed from different disciplines such as sociology, philosophy, or law. These works have been taken as reference for the definition of norms that control agent societies. This section contains an overview of these background works.

2.2.1 Sociological Approach to Norms

Sociology is the social science that is focused on the study of both society and social phenomenon, i.e., social action, social relationships, and social groups. It studies how the organizations and institutions that make up the social structure are created, maintained,or changed. It also studies how these social structures (i.e., institutions and organizations) affect individual and social behaviour, and how social structures are adapted as a consequence of the social activities.

Therefore, both organization and institution concepts have been defined by sociology as abstractions in order to analyse the way in which human beings cooperate and coordinate themselves. These abstractions have also been employed in the MAS field for modelling agent societies. Thus, an *organization* is understood to be a permanent arrangement of elements. An organization consists of a set of individuals who carry out some specific and differentiated activities or tasks. Moreover, they are structured following some patterns or rules that allow them to achieve the organizational goals [Etz64, Sco02]. *Institutions* are structures and mechanisms of social order and cooperation that govern the behaviour of a set of individuals. The essential role of human institutions is to create new types of *power* relationships. Power is related to terms such as: rights, responsibilities, duties, etc. Therefore, powers are also known as *deontic powers* (norms) [Sea05]. The definition of deontic powers differentiates human societies from animal societies. Therefore, human societies are identified with a social purpose and permanence, transcending individual human lives and intentions, and with the making and enforcing of rules governing cooperative human behaviour. Institutions are a central concern for law, the formal regime for political rule-making and enforcement. Institutions are "collectively accepted systems of rules that enable us to create institutional facts". These *institutional facts* are those facts that occur as a consequence of collective acceptance and recognition. For example a piece of paper will only be money as long as the members of the society believe that it is so. The

existence of money is an institutional fact. A deeper analysis of the difference between the organization and institution concepts is outside the scope of this work and has been given a lot of attention in the existing literature [Sco95]. In this thesis, the main difference between these two social structures lies in the legal features of institutions, which explicitly create new states of affairs (i.e., institutional facts and norms). Thus, institutions are a concrete type of human organization characterised by the existence of deontic powers.

As stated above, social structures have an effect on human behaviour. Norms are one of the most important phenomena that influence both individual and social behaviour. A norm is defined in [Gib65] as an entity composed of three parts: (i) it is a "collective evaluation of behaviour in terms of what it ought to be" or what it ought not to be; (ii) it is also a "collective expectation as to what that behaviour will be"; and (iii) it may or may not "include particular reactions to behaviour, including attempts to apply sanctions or otherwise induce a particular kind of conduct". Thus, normative features are divided into two sets [Gib65]: definitional features, which are established in the norm definition (i.e., the expected behaviour and the collective evaluation); and contingent attributes, which may or not occur (i.e., the application of sanctions or rewards). This distinction among definitional and contingent features allows the definition of more complex and concrete types of norms such as moral norms, rules, laws, and so on. *Social norms* [Mor56] are the type of norm that has received the most attention from sociology. The work described in [Els89] characterizes social norms as follows:

> "For norms to be social, they must be shared by other people and partly sustained by their approval and disapproval. They are also sustained by the feelings of embarrassment, anxiety, guilt and shame that a person suffers at the prospect of violating them. A person obeying a norm may also be propelled by positive emotions, like anger and indignation."

According to this, the emotional and social dimensions of norms are the key factors that allow the distinction among social norms and other kinds of norms such as private ones. For example, private norms can be sustained by feelings of anxiety and guilt, but they are not shared by society.

One of the most cited works on the classification of social norms is [Tuo95]. This work classifies social norms into *r-norms* and *s-norms*. The former are norms, or *rules*, which have been promulgated by an organization authority or the institution itself. These rules are contained in jurisprudence documents such as regulations. Thus, their violation is considered to be an illicit

act and entails sanctions or punishments. In contrast, *s-norms* are those norms that emerge from social *conventions*. The s-norms indicate the established and approved ways of doing things, of dressing, of speaking and of appearance. They vary and evolve, not only through time, but also from one age group to another and among social classes and social groups. Their violation does not imply an institutional sanction or punishment; however, by ignoring the social norms, one risks becoming unacceptable, unpopular or even an outcast from a group. Norms of this kind tend to be tacitly established and maintained through body language and non-verbal communication between people in their normal social discourse.

In summary, sociology has defined criteria for classifying norms into two main categories; i.e., private and social norms, according to the norm scope. In addition, social norms are divided into r-norms (we will refer to norms of this type as *institutional* norms) and s-norms. Besides the definition and characterization of the different norm types, sociology has also dealt with a justification for the existence of norms and explained motivations for norm adherence [Els89]. The main conclusion of this work is that norms are substantiated by rational motivations such as self-interest motivations (e.g., fear of sanctions, interest in rewards) and common interests. Moreover, norms are maintained by emotions such as anxiety and shame, honour, and envy, among others.

2.2.2 Philosophical Approach to Norms

Philosophy is a discipline that attempts to understand things such as the nature of reality and existence, the use and limits of knowledge, and the principles that govern and influence moral judgement[1]. *Deontic logic* is a logic system used for the formal analysis of norms and propositions about norms. It can be defined as the study of those sentences that are formed by *normative expressions* (e.g., "obligation", "permission", and "prohibition"). The "deontic" term is derived from the ancient Greek term *don* which means "as it should be" or "duly"[McN10].

Leibniz is the precursor of deontic logics. In 1671, he pointed out the analogy between the normative concepts "fair", "unfair", and "optional" with the athletic model concepts "necessary", "possible", and "impossible". However, the first philosopher that attempted to build a formal theory dealing with normative concepts is Mally [Mal71] (first published in 1926) . This work presents an axiomatic system for covering the notion of *ought*. This system is unsatisfactory, since it allowed absurd theorems to be proved. However, it is the first logic system

[1]http://dictionary.cambridge.org/

that included normative concepts. The most relevant work on deontic logics is the contribution of von Wright [vW57] (first published in 1951). He proposed the first viable system for deontic logics. This approach is also based on the similarity among deontic notions of *obligation* and *permission* and the model notions of *necessity* and *possibility*. Therefore, deontic logic is interpreted as a branch of modal logics. This work confronts the definition of a logic system for norms from a syntactic or axiomatic perspective. The later works of Kanger [Kan71] and Kripke [Kri63] give a semantic interpretation to the deontic logic system. As argued by von Wright, there are a large number of outstanding problems or pradoxes in deontic logic. Further details about the use of deontic logics for representing agent norms are contained in Section 2.4.1.

2.2.3 Legal Approaches

Law is a system for organizing human societies. Basically, it is composed of *institutional* norms, which regulate social coexistence and allow interpersonal conflicts to be solved. In addition, the legal system employs institutions as structures for enforcing norms.

One of the most relevant works in the law literature is the book by [AB71]. It defines a *normative system* as a set of statements in which there are some *normative statements* or norms. Normative statements are those that define an action as obligatory, forbidden, or permitted. In addition, this definition of normative system allows the system to enunciate non-normative statements, which, for example, are definitions of the terminology employed in norms. This work provides a deeper analysis of normative systems and their properties. Specifically, the structural properties of normative systems are:

- *Completeness.* This property characterises those normative systems that contain enough norms to solve each possible situation or case.

- *Independence.* An independent normative system is one that does not contain redundant norms. A norm is defined informally as redundant when it is unnecessary; i.e., the normative system without this norm remains equivalent to the original one.

- *Coherence.* Informally, a normative system is defined as incoherent when it contains two or more contradictory norms. Contradictory norms are those that deontic propositions that are logically incoherent for the same case; e.g., norms that define something as forbidden and permitted in the same situation.

2.2.4 Artificial Intelligence & Law

The first works dealing with norms from the perspective of the *Artificial Intelligence* (AI) field attempted to model legal research, reasoning, and argumentation in a computable way to allow legal systems to be automatically evaluated and analysed [RAL03]. AI & Law is a classic field of AI that has dealt with legal issues that are relevant for MAS, such as the logic for formalising deontic propositions and normative relationships (Section 2.4), the dynamics of normative systems (Section 2.7), normative reasoning and argumentation (Section 2.6), and so on. As a consequence, initial works on norms for regulating MAS have taken their inspiration from the AI & Law field, whereas the AI & Law discipline has moved toward the MAS area looking for new domains of application [RS08].

2.3 Norm Definition in Multi-Agent Systems

The norm concept has been ambiguously employed by different disciplines as a synonym of law, rule, guideline, criterion, social expectation, and imperative. Similarly, normative systems have been given different definitions. Relevant proposals on the definition of both the norm concept and normative systems in the MAS field are described in this section.

2.3.1 Norm Definition

Norms have been proposed in MAS research as formal specifications of deontic statement aimed at regulating the life of software agents and the interactions among them [RS08]. More specifically, norms have been proposed to deal with coordination issues [LyLLd02], to deal with security issues in MAS [UBJ+03], to model legal issues in electronic institutions and electronic commerce [GCNRA05], and to model MAS organizations [DVSD04].

A *normative multi-agent system* (NMAS) combines models for normative systems with models for multi-agent systems. Therefore, NMAS have been defined as the research field formed by the intersection between normative system theory and the MAS area [BvdTV07]. *Normative systems* have been redefined in computer science as "systems in the behaviour of which norms play a role and which need normative concepts in order to be described or specified" [MWD98]. Next, works on the classification of those norms that are used in NMAS are described.

2.3.1.1 Norm Typology

A classification of norms according to their purpose is proposed in [BvdT08]. This classification takes well-known philosophical works as a reference [Gib65, Sea69, The02] and divides norms into substantive and procedural norms.

- *Substantive Norms.* They define the legal relationships among the members of the society and the normative system itself in terms of regulative and constitutive norms.

  - *Regulative Norms.* They describe the ideal behaviour and varying degrees of sub-ideal behaviour by means of obligations, prohibitions, and permissions [BvdT08, LyLL03]. These norms regulate activities that can exist independently of the norm. An example of a regulative norm is "It is forbidden for students to speak in an exam".

  - *Constitutive Norms.* Norms of this type give an abstract meaning to facts, environmental elements, etc. . They introduce new classifications of abstract facts and entities, named institutional facts (see Section 2.2.1). They provide an abstraction mechanism, namely to define the ontology used for describing the behaviour of the system [BvdT04a]. Thus, constitutive norms also describe the legal consequences of actions in the normative system. Therefore, *legislative* norms [LyLL03], which are metanorms that define how the normative system is modified by agents, are also constitutive. Constitutive norms are defined by means of *count-as* conditionals [Sea05]. These conditionals are expressions such as $X\ count-as\ Y\ in\ C$, which represents that the basic or brute fact X can be redefined as the institutional fact Y in context C. An example of a constitutive norm is "A situation in which students are asked to solve some exercises privately counts as an exam".

- *Procedural Norms:* these norms are an instrumental approach; i.e., they are aimed at achieving social order specified in terms of substantive norms. In this sense, there is no logical connection between a regulative norm and a sanction or reward. Therefore, *procedural norms* define a practical connection between a regulation and its consequences [BvdT08]. Procedural norms define how rewards, costs and risks are allocated within a social system. Thus, they are also known as *enforcement* norms [LyLL03]. An example of a procedural norm is "Teachers are obliged to fail students that have violated the

speaking prohibition"; which obliges teachers to enforce the speaking prohibition by failing dishonest students as sanction.

This classification divides norms taking into account their purpose: to define new classifications of facts (constitutive), to define the ideal behaviour of the system (regulative), or to connect ideal behaviours to enforcement mechanisms (procedural).

2.3.1.2 Norm Levels

There is a classification of norms into three levels, according to the norm scope [Dig99]. The norm scope defines the ambit of a norm; i.e., the range of agents affected by the norm.

1. *Social level.* This is the highest level of norms. It is made up of norms that govern the coordination of individuals in a society (see Section 2.2.1). As previously argued, this set of norms is formed by *institutional norms* and *conventions.*

   Institutional norms are explicitly promulgated by the institution (or a representative); their violation is considered as an illicit act and implies sanctions. Thus there is a representative entity which has been empowered for sanctioning agents that violate the norms. Institutional norms are in force until they are abolished by the institution (or a representative). For example, "Citizens are forbidden to kill people" is a well-known example of an institutional norm.

   Conventions or *social norms* emerge from agent behaviour as a macro-level effect derived from an interaction among agents. These norms have not been defined explicitly and, logically, they are not enforced by an entity which represents the institution. Therefore, these norms do not define sanctions and rewards for persuading agents to respect them. However, they are enforced by social mechanisms such as ostracism, recrimination, etc. "It is obliged to be smartly dressed at a gala dinner" is an example of a social convention.

2. *Interaction level.* This is the intermediate level of norms. Both legal contracts and informal agreements between entities belong to this level. An *agreement* is a decision or arrangement between two or more groups or people, whereas a *contract* is a legal document that states and explains a formal agreement[2]. Norms of this type are created explicitly for a limited period of time as a consequence of an interaction among individuals

[2]http://dictionary.cambridge.org/

or groups of individuals acting as a single entity. They are also based on the notion of obligation, prohibition and permission. Each interaction norm indicates how it arises, how it is fulfilled, and what happens if it is fulfilled or not. Thus, these norms normally include sanctions and rewards. Agents affected by the an interaction norm are usually responsible for monitoring its fulfilment. An example of an interaction norm is "I am permitted to use my father's car since he agrees with it".

3. *Private Level.* This is the lowest level of norms. Private norms are norms that are self-imposed. Private norms ensure agent autonomy. If agents' behaviour were only determined by norms belonging to higher levels, their behaviours would be completely directed by external norms. In this sense, private norms are created inside agents' minds and they are accepted as principles. Private norms are not enforced by sanctions and rewards. These internal norms may be created as a result of the internalization of an interaction or social norm. In this sense, "I must be polite" is an example of a private norm that has been created in conformity with good manners.

Examples of regulative norms belonging to each one of these levels have been provided. In addition, constitutive and procedural norms can be defined. For example, two agents may reach a contract for purchasing oranges. Thus, in the context of this contract, a constitutive norm defines what is considered as high quality oranges (e.g., "high quality oranges are those that have a minimum size"). Moreover, a procedural norm may define what would happen if the delivered oranges do not respect the minimum size restriction (e.g., "in the case of contract violation, due to the small size of oranges, the seller is permitted to reduce the agreed price").

2.3.2 Normative Multi-Agent Systems

NMAS provide a promising model for human and artificial agent coordination because they integrate norms and individual intelligence. They are a clear example of the use of sociological theories in multi-agent systems, and therefore of the relation between agent theory (both multi-agent systems and autonomous agents) and the social sciences (sociology, philosophy, economics, legal science, etc.) [BvdTV08a]. There have been different definitions given over time. In 2005 NMAS were defined as *"MAS together with normative systems in which agents on the one hand can decide whether to follow the explicitly represented norms, and on the other hand the normative systems specify how and in which extent the agents can modify the norms"*

[BvdTV06]. More recently, in 2007, NMAS were defined as follows *"a MAS organized by means of mechanisms to represent, communicate, distribute, detect, create, modify, and enforce norms, and mechanisms to deliberate about norms and detect norm violation and fulfilment"* [BvdTV08a]. The main distinction between these two definitions is that the former focuses on the representation of norms, whereas the latter is more related to the mechanisms employed for organizing MAS. In this sense, the interest in NMAS has evolved from a static legalistic definition of norms into a more dynamic interactionist perspective. Thus, norms have been interpreted from two different perspectives [BvdTV08b]:

- The *legalistic* perspective is a top-down view that considers the normative system as an instrument for regulating the emerging behaviour of Open Systems [AP01], in which heterogeneous agents can participate. Norms set up the basis for agent interactions. Norms are explicitly created by the system designer or a representative agent. However, norms are not imposed on agents; on the contrary, agents are persuaded to behave according to the norms by means of sanctions or rewards.

- The *interactionist* perspective is a bottom-up approach that considers norms as conventions that emerge from agent interactions. Thus, norms must be communicated and spread in the society. However, their enforcement cannot be delegated to the MAS infrastructure, so mechanisms for a social enforcement of norms are necessary. This perspective is more related to the notion of social norm.

According to this evolution in the definition of NMAS, five levels in the development of NMAS have been proposed in [BvdTV08b]. However, only the first four levels are considered here. The last level is related to machine ethics [AA07], which is closer to ethical theory. Table 2.1 illustrates the different levels in the development of NMAS. Level 1 is composed of closed systems in which norms are defined off-line by the system designer and imposed (hard-wired) on agents. Thus, agents are not autonomous for deciding whether they observe norms. At level 2, norms are explicitly represented, and agents can be aware of them. Therefore, agents are autonomous to follow norms. As a consequence, mechanisms for enforcing norms are needed. At level 3, norms are not only explicitly represented, but they can also be manipulated by agents (i.e., agents can add or remove norms). Thus, norms can be dynamically adapted to the requirements of each particular situation. The development of NMAS has reached this third level. Nowadays, the NMAS area is moving to the 4th level. This higher level corresponds

to the interactionist view, in which agent interactions are the base for norms. Thus, norms are emergent regularities of behaviour that are sustained thanks to social mechanisms such as blame and exclusion of non-conforming agents. Therefore, research focus has evolved from works aimed at addressing logical and representational issues to issues such as: agent decision making, norm dynamics, legislator roles, etc. [BPvdT09b].

2.4 Norm Representation

In order to employ norms to achieve coordination and cooperation inside dynamic societies formed by heterogeneous agents, a formal model of norms is needed. This model should represent the *prescriptive* and *descriptive* dimension of norms [vdTT01]. *Norms* have a prescriptive meaning, i.e., they describe the desired behaviour. Therefore, they can be adopted or not, and respected or not, but they cannot be described as true or false. For example "forbidden to kill" is an example of norm prescription. But norms also have a descriptive meaning, i.e., they describe the norms that govern a society and the normative relationships that exist among the society members. Following with the previous example, "in our society it is forbidden to kill" is a description of the fact that the normative system defined by our society contains a prohibition to kill. Thus, this section describes the main works on the formalization of both the prescriptive and descriptive dimensions of norms.

2.4.1 Deontic Logic: Logic of Norms

Regarding the formalization of norms, norms define the rights and duties of the society members in terms of permissions, prohibitions, and obligations [vdTT99b]. Next, the main works on the representation of normative prescriptions will be introduced. All of these proposals are based on the deontic logic, whose fundamentals are described below.

2.4.1.1 Standard Deontic Logic (SDL)

The most well-known system of deontic logic is the *Standard Deontic Logic* (SDL) [vW57]. Basically it consists of a language or classic propositional logic, the negation ($\neg$) and consequence ($\rightarrow$) operators and the *deontic operators* (**O** for representing obligations, **P** for permissions and **F** for prohibitions). SDL is axiomatised as follows:

Level	Name	Features
1	**Off-line Design of Norms**	Norms are hard-wired on agents and are automatically enforced Agents cannot organize themselves by means of norms
2	**Norm Representation**	Norms are explicitly represented Norms can be used in agent communication and negotiation
3	**Norm Manipulation**	Norms are manipulated by agents
4	**Social Reality**	Norms are emergent regularities of behaviour There is not any enforcement system Norms are respected by social blame and by the expulsion of non-conforming agents

Table 2.1: Levels in the development of NMAS [BvdTV08b].

All tautologies of wffs [3]	(Taut)
$\mathbf{O}(p \rightarrow q) \rightarrow (\mathbf{O}p \rightarrow \mathbf{O}q)$	(O-K)
$\mathbf{O}p \rightarrow \neg\mathbf{O}\neg p$	(O-D)
$\mathbf{P}p \leftrightarrow \neg\mathbf{O}\neg p$	(P)
$\mathbf{F}p \leftrightarrow \mathbf{O}\neg p$	(F)
$p, p \rightarrow q \vdash q$	(MP)
$p \vdash \mathbf{O}p$	(O-NEC)

Each one of the deontic principles has been questioned; in fact, there are several works on the paradoxes and inconsistencies of deontic logic [HPvdT07]. Works aimed at avoiding and solving some of these paradoxes are described in Sections 2.4.1.2, 2.4.1.3 and 2.4.1.4. Semantics of SDL, given by means of a possible worlds model (i.e., a Kripke structure [Kri63]), has not been included here due to space limitations.

In [And58] the reduction (known as *Andersonian reduction*) of deontic logic into a modal logic is proposed. This interpretation of deontic logic has been widely employed for modelling normative systems. Basically, a *modal logic* [Che80] is a propositional logic that is extended with the modal operators $\Box$ and $\diamond$, which represent *necessity* and *possibility*, respectively. Anderson's proposal consists of extending classic modal logic with a deontic propositional constant (d), which represents the fact that all norms are satisfied. Therefore, deontic operators are defined as follows:

$$\mathbf{O}p \Leftrightarrow_{def} \Box(d \rightarrow p)$$
$$\mathbf{P}p \Leftrightarrow_{def} \Diamond(d \wedge p)$$
$$\mathbf{F}p \Leftrightarrow_{def} \Box(d \rightarrow \neg p)$$

Thus, p is *obliged* ($\mathbf{O}p$) iff p is entailed (*necessitated*) by all normative demands being met (d); p is *permitted* ($\mathbf{P}p$) iff it is compatible (*possible*) with all obligatory states of affairs (d); finally, p is *forbidden* ($\mathbf{F}p$) iff it is incompatible with all normative demands. For example, a prohibition norm about a proposition a ($\mathbf{F}a$ is defined in modal logic as $\Box(d \rightarrow \neg a)$. This means that in situations where the norms are satisfied (d), the proposition a does not hold.

Frequently, the norms that regulate agent behaviours depend on past actions and events. In order to represent these relationships, deontic logic has been extended with operators belonging to temporal [DMWK96] and dynamic logics [Mey87].

[3] *well-formed formulas*

These first works on the development of logics for norms gave rise to some problems, as illustrated in [Chi63]. Modern works on deontic logic [Han69] not only classify words (facts or actions) as good or bad (legal or illegal); but they propose the employment of a betterness relation among words or situations. Following this intuition, the Preference-Based Deontic Logic, which is explaied below, uses an order relationship for classifying states according to a preference function.

2.4.1.2 Preference-Based Deontic Logic (PDL)

As mentioned in Section 2.4.1.1, several paradoxes arise in SDL and relate logic systems [FH71]. For example, the well-known Ross' Paradox [Ros44] consists of:

$$\mathbf{O}p \rightarrow \mathbf{O}(p \vee q)$$

In [McN10], a specific example of this paradox is provided:

$\mathbf{O}p$	*it is obligatory for the letter to be mailed*	(1)
$\mathbf{O}(p \vee q)$	*it is obligatory for the letter to be mailed or for the letter to be burned*	(2)

We have $p \rightarrow p \vee q$ by disjunction introduction. So, we have $\mathbf{O}(p \rightarrow p \vee q)$ by axiom (O-NEC) (see the axiomatization of SDL in Section 2.4.1.1) which can be written as $\mathbf{O}p \rightarrow \mathbf{O}(p \vee q)$ by (O-K). Thus, (2) follows from (1) by (MP). However, it seems rather odd to say that an obligation to mail the letter entails an obligation that can be fulfilled by burning the letter (something that is presumably forbidden).

To avoid paradoxes of SDL, [Han90] provides a new possible world semantics for deontic logic. This formalism, known as *Preference-based Deontic Logic* (PDL) is based on a preference logic, which is a logic system that defines a preference relationship that mainly defines an action (or a set of actions) as preferable or indifferent with respect to other actions. Taking the implicit ordination of actions provided by the preference relationship, the PDL formalism adds the notion of normative predicates that express prescription or prohibitions of different degrees. As shown in [Han90], theorems derived from PDL do not present the paradoxical nature as SDL theorems. In addition, plausible axioms of the SDL are also present in PDL formalization.

2.4.1.3 Dyadic Deontic Logic (DDL)

In [Lew74], the *Dyadic Deontic Logic* (DDL) is proposed. The main difference among SDL and DDL is that deontic operators are dyadic deontic logics that contain binary deontic operators: $\mathbf{O}(A \mid B)$ means it is obligatory that A, given B; and $\mathbf{P}(A \mid B)$ means it is permissible that A, given B. This logic has been proposed in order to overcome Forrester's paradox [For84]:

$\mathbf{O}\neg m$	*it ought to be that Smith not murder Jones.*	(1)
$m \rightarrow \mathbf{O}g$	*if Smith murders Jones, Smith ought to murder Jones gently*	(2)
$g \rightarrow m$	*gently murdering implies murdering*	(3)
m	*Smith murders Jones*	(4)

From (2) and (4), by modus ponens, we get $\mathbf{O}g$. Then from (3) by (MP) (see the axiomatization of SDL in Section 2.4.1.1) $\mathbf{O}g \rightarrow \mathbf{O}m$ is obtained. From these two, by modus ponens, $\mathbf{O}m$ is obtained, which is inconsistent with (1). This paradox is an example of contrary-to-duty structures [Chi63], which are situations in which there is a primary obligation and a secondary obligation that comes into effect when the primary obligation is violated. The representation of these deontic statements is the source of one of the main deontic logic paradoxes.

The cause of these paradoxes is that deontic logic cannot be subsumed under normal modal logics. Thus, contrary-to-duty obligations cannot be faithfully expressed in SDL making use of a unary deontic operator and a material conditional. As a response to this issue, DDL supposes that any system of norms induces a ranking on possible contexts or situations with respect to the extent to which the histories comply with norms. The highest ranking possible contexts are those in which no norm is violated. As one descends the ranking, more and/or more serious violations occur. This allows for the evaluation of conditional obligation sentences. $\mathbf{O}(A \mid B)$ holds iff A holds at all the highest ranked histories at which B holds.

2.4.1.4 Defeasible Deontic Logic

Also in response to contrary-to-duty paradoxes of SDL, in [Nut97] *Defeasible Deontic Logic* was proposed. The main idea of this proposal is to combine deontic logics (SDL or DDL) with defeasible logic. Defeasible logic [Nut03] is the logic of default assumptions. It is a non-monotonic logic in which there are: rules that specify that a fact is always a consequence of another; and *defeasible* rules that specify that a fact is typically a consequence of another.

The main intuition of defeasible deontic logic [Nut97] is to write conditional obligations, such as contrary-to-duty ones, as defeasible rules (known as defeasible deontic rules). In [vdTT97], three types of defeasibility in deontic logics are analysed: *factual* defeasibility, which models the overshadowing of an obligation by a violation fact (this issue is related to the verification and detection of norm violations discussed in Section 2.5.3); *strong overridden* defeasibility models the overshadowing of an obligation by more specific obligations (Section 2.6.1.3 focuses on this issue, considering conflicts and inconsistencies among norms); and *weak overridden* defeasibility models *prima facie* obligations, which are obligations that can be overshadowed but not cancelled.

Another relevant proposal on the relationship among defeasible logic, deontic logic, and agency was made by Governatori et al. in [GR04, GRS05]. In [GR04], a defeasible multi-modal logic arising from the combination of agency, intentions, and obligations is proposed. This proposal takes as a reference the Nute's general definition of defeasibility [Nut03] and defeasible deontic logic [Nut97]. More recently, this proposal was extended in [GRS05] with the consideration of temporal considerations. In particular, this work concerns the temporal and dynamic treatment of deontic statements.

2.4.2 Input/Output Logic

As previously argued, it makes no sense to judge norms as true or false. The *Input/Output* (I/O) logic [MvdT00] has been developed to formalize systems of norms that do not bear truth values.

According to the I/O logic, norms are modelled as ordered pairs (a, x), where a is a propositional input that represents some condition; and x is a propositional output that represents what the norm defines as mandatory. Thus, norms are not used using truth-functional connectives. Prohibitions can be defined similarly in I/O logic as $(a, \neg x)$, meaning that,in a situation a, the negation of proposition x is forbidden.

Of special interest is the formalization of permissions. In [MvdT03], an in-depth analysis of the permission from the perspective of I/O logic is made. In particular I/O logic makes a clear distinction among *negative* and *positive* permissions. The former is a negation of an obligation. Two kinds of positive conditional permissions have been identified. Specifically, *static positive permissions* define that a proposition is permitted guiding citizens in the assessment of actions, so they are seen as weakened obligations. On the other hand, *dynamic positive permissions*

guide the legislator by describing the limits on the prohibitions that may be introduced into a set of norms.

In [BvdT03], the above-mentioned types of permissions are taken into account in order to study how permissions can dynamically change a normative system by adding exceptions to obligations, providing an explicit representation of what is permitted and allowing the definition of a hierarchy of authorities. With regard to this last question, higher level authorities can define dynamic positive permissions that determine the way in which lower level authorities issue norms. The question of norm change from the perspective of I/O has been tackled by later works, which will be explained in Section 2.7.1.2.

2.4.3 Commitments

The work in [Sin99] proposes one of the first models of agent *commitments*. A commitment is defined as a set of conditions that should be satisfied as a consequence of an agent interaction. In addition, a set of operators for working with commitments are also defined. An interesting contribution of this work is the presentation of normative concepts such as obligations (explained in Section 2.4.1.1), conventions (described in Section 2.7.2), and rights (see Section 2.4.1) in terms of commitments. As pointed out by this article, there is a close relationship between commitments and illocutions, which are acts of speaking which in themselves effect or constitute the intended actions. Semantics for *Agent Communication Languages* (ACLs) is provided in [Sin00]. This semantics is not focused on the agent mental states but on the interactions among agents. It is based on the commitments that are created implicitly by the illocutions. This model of commitment has been used by later works on the formalization of protocols based on commitments [YS02], on the verification of compliance with these protocols [VS99], and on the adaptation of protocols according to different contexts or situations via transformations [CS06], among others.

To provide the meaning of ACLs, an operational definition of a commitment-based semantics for communicative acts is provided in [FC02]. This proposal provides a complete account of how different types of speech acts can be defined in terms of operations on commitments. In addition, a commitment-based analysis of directive speech acts (e.g., a request for a commitment from a second party) is provided. The authors model the life cycle of commitments in the system through update rules. Based on these update rules, a commitment can either be fulfilled, violated, or cancelled.

In [BMMCd04], the authors propose a formal model of commitment. This model consists of three main concepts: commitments, actions, and arguments to support these actions. All of these concepts are formally described by means of *Computational Tree Logic* (CTL*) [Eme90] and dynamic logic [Har84].

2.4.4 Social Law

The *social law* paradigm was proposed in [ST92b] by Shoham and Tennenholtz. A social law is a set of functions that restrict the permitted actions for an agent at each moment. From this perspective, the fundamental problem of designing a MAS consists in defining the set of constraints (social laws) over agent actions [MT95]. These restrictions must lead to a system in which each agent can achieve its design goals, reaching an appropriate balance among the conflicting goals of agents. This proposal assumes that the social laws are hard constraints that are defined *off-line* (see Section 2.7.1.1 for more details about the problem of creating social laws). This is in fact the main drawback of this model. It is unsuitable for open systems, in which the adaptation, modification and even the violation of norms may be essential.

The social law proposal has been the basis for later works such as [vdHRW07] and [BvdT05a]. In [vdHRW07], the authors provide an alternative formalization of the social law model by means of the *Alternating-Time Temporal Logic* [AHK02]. Therefore, the problem of creating social laws can be implemented as a model checking problem without adding extra complexity to the corresponding problem in the original framework of Shoham and Tennenholtz. In [BvdT05a], authors focus on the definition of control mechanisms for monitoring and enforcing social laws.

2.4.5 Normative Positions

The theory of *Normative Positions* was initially developed in [Kan72] and [Lin77]. These works define the term *normative position* as the set of normative relationships that can be defined in an agent society (e.g., right, duty etc.). The *language of normative positions* [Kan72] consists of a First-Order Logic increased with the modal expression for obligations (O) and the modal operator (Do) representing actions carried out by agents. The main idea of this theory is that, given a certain assumption, e.g., $O(Do(a, F))$; which means agent a is obliged to perform action F, the theory generates the normative positions (i.e., the permissions and obligations) of another

agent b which are consistent with the given assumption. These normative positions allow more complex normative relationships such as right, duty, authorization, etc. to be defined. A study of these different relationships or normative positions is made in [Lin77] . Making use of the works cited above, a refinement of this theory is made in [Ser98, Ser01], taking into consideration its application in the computer science field. The initial works on normative positions are extended to consider relationships among more than two agents. These methods for generating and calculating the normative positions are general enough to support different logic formalisms for representing deontic relationships and actions. In addition, an algorithm that carries out the inferential process for determining the normative positions among a set of agents in a certain situation has also been proposed [Ser98, Ser01]. Specifically, the problem of generating normative positions has been modelled as a graph colouring problem. These graphs represent *state transition systems*. The $nC+$ language for representing normative systems as transition systems is presented in [SC06]. This approach consists of labelling the states and transitions as legal or illegal (red or green). Taking this definition of *labelled transition system* as a basis, the problem of generating normative positions consists of labelling (colouring) the states of the graph representing normative states. One of the main drawbacks of this proposal is the lack of temporal notions for representing temporal constraints over deontic and action formulae. Also, this formalism does not allow a representation of *power* relationships and power positions. The notion of power in normative systems is explained below.

2.4.6 Power in Normative Systems

The notion of *power* (mentioned in Section 2.2.1) has a close relationship with *constitutive norms* (see Section 2.3.1.1). In particular, constitutive norms define the *count-as* relationship which defines how the institutional reality is built in terms of actions or state of affairs occurring in the real world. The relationship between power and constitutive norms has been studied by Jones and Sergot in [JS96]. In this work, they propose the formalization of the *count-as* relationship in any action logic by means of the $\Rightarrow_s$ operator. The expression $X \Rightarrow_s F$ means that within the institution or context s occurrence of X *count-as* Y. Specifically, Jones and Sergot make use of this operator in the definition of *institutional power*, which they define as:

> "the constraints whereby an institution makes particular kinds of acts or particular kinds of states of affairs count as sufficient conditions for guaranteeing the applicability of particular classificatory categories and these classifications when made

> often carry with them certain kinds of normative consequences concerning rights and duties"

The distinction between power and permissions is that if an agent is not empowered to perform an action that is affected by a *count-as* relationship, then this *count-as* relationship will not be applied and the action will not have institutional consequences. On the contrary, if the agent is empowered but it is forbidden to perform this action, then the institutional effects will take place and it will be considered as a violation.

This notion of power has been employed in later works on the definition of institutions for agent societies. For example, in [CFV02, FVC07], Sergot and Jones' notion of *institutional power* is represented under the concept of 'authorization'. In this work, agents are authorised to change the institutional state when they are given official permission to do it. Thus, authorizations are necessary conditions for the valid performance of institutional actions. *Institutional actions* are a particular type of actions whose effects are to change institutional facts (see Section 2.2.1) which exist only thanks to common agreements.

2.4.7 Norms and Time Considerations

For the time factor, the work contained in [ADM07] proposes the usage of the *Linear-Time Logic* (LTL) [Kro87] for expressing agent norms. More specifically, deontic operators of obligation and prohibition are defined by means of LTL formulas. This proposal is completed with a mechanism that allows agents to generate actuation protocols according to the institution norms [ADM07]. Another interesting proposal of logic for representing norms is the *Normative Temporal Logic* (NTL) proposed in [ÅvdHRA+07]. The NTL is based on CTL [Eme90], but the universal and existential operators have been replaced by deontic operators of obligation and permission. Its semantics is given in a Kripke style, i.e., a possible worlds model in which transitions have been labelled as legal or illegal. The main advantage of this proposal is that several works on verifying properties and reasoning about normative systems have been done in [ÅvdHW07] and [ÅvdHW08]. These works are described later in Section 2.6.1.3.

Finally, the term *social expectation* [Cra06] has been employed to cover any present or future restriction resulting from a set of rules that represent the social convictions. $hyMITL^{\pm}$ [Cra07] is a variant of temporal logic that allows the representation of expectations as conditional rules defined over past and current observations, whose consequences impose restrictions on future

states. In addition, an algorithm that allows both agents and the system designer to reason about expectations has been developed [CW07].

2.4.8 Open Issues for a Logic of Normative Systems

As explained above, open NMAS require a logic formalism to explicitly represent both norms and normative systems. Thus, the MAS field must pay more attention to recent works on deontic logics such as: the logic of imperatives [Han04] or normative systems [MWD98]. Regarding the expressiveness requirements, there is a lack of logics for norms which are aimed at groups. The new proposals must provide ways of formalising *group responsibilities, rights and duties*; and how they are distributed and shared by the members of the group. Another issue that must be considered by future works is the consideration of *uncertainty* of the environment in which agents' interactions take place. Specifically, to consider norms as more than just simple formulae that hold or not hold in a specific moment, there is a need for a more elaborate representation in which there is a possibility of uncertain interpretation of norms. This raises the need for more complex procedures to detect and reason about *norm compliance*. This last issue will be described in Section 2.5.

As previously pointed out, these logical formalisms for open NMAS must allow these types of systems to be reasoned about, which implies the study of adequacy and consistency properties, as well as the individual reasoning of agents whose interactions are regulated by norms. The following section deals with the normative reasoning problem.

2.5 Norm Implementation

Most of these proposals deal with norms from a theoretical perspective. However, several works on norms from an operational point of view have recently emerged. These approaches are focused on giving a computational interpretation to norms for use in the design and execution of MAS applications. This section is not focused on the implementation of norms as agents' mental attitudes (this issue is covered by Section 2.6.1). This section illustrates how norms can be implemented inside NMAS from an institutional perspective. In Open NMAS, internal states of agents are not accessible. Therefore, norms cannot be imposed as agent beliefs or goals, but they have to be implemented in the society by means of control mechanisms.

2.5.1 Normative Language

The explicit representation of norms in NMAS allows norm-aware agents to be informed about the norms that are in force at a specific moment. Thus, agents will be able to modify their behaviour accordingly. In the existing literature about norm implementation, there are several proposals on normative languages. Mainly, these languages allow the definition of deontic constraints that restrict the potential excesses of agents' autonomous behaviours. This section describes proposals on normative languages that are close to the implementation of norms (normative languages related to logic issues were described in Section 2.4).

In, [VSAD04] Vázquez-Salceda et al. propose a general purpose normative language (described in Table 2.2, first row). In this language, a norm mainly specifies a deontic control over an agent and a situation. These situations can be defined over a state condition or an action. In addition, norms may have conditions for their activation and temporal constraints. This language has been employed as a reference for other proposals on normative languages. For example, in [GCNRA05], an extension of this proposal to define norms in *Electronic Institutions* (EI) is proposed (see Table 2.2, second row). EIs represent a way to implement interaction conventions for agents who can establish commitments in open environments. Thus, valid agent actions inside an EI are communication acts or illocutions (i.e., actions that are performed by saying something) [Est02]. Therefore, norms in EI are defined over dialogical actions (i.e., the pronunciation of illocutions). In order to provide support to non-dialogical actions, an extension of the normative language for EI has been proposed in [dS07].

The *Contract*[4] project has produced a new language for the expression of contracts between web services (see Table 2.2, third row). It takes into account computational issues of reasoning over contracts and how the properties of contract systems can be verified. Specifically, a contract contains a name, starting and ending dates, a contextualization, definitions, and clauses. These clauses are expressed as norms that are formed by [OPVS$^+$09]: a *type* identifier, stating whether the norm is an obligation or a permission; an *activation condition*, stating when the norm must be instantiated; a *normative goal* or state (condition) used to identify when the norm is violated (in the case of obligations and prohibitions), or what the agent is permitted to do; an *expiration condition*, used to determine when the norm no longer affects the agent; and a *target*, identifying which agents the norm affects.

Another interesting proposal of normative language is the one described in [ACBJ08] (see

[4]http://www.ist-contract.org

Table 2.2, fourth row). This work proposes a normative language for controlling agent access to services. Thus, it describes agents' permissions, obligations, and prohibitions for requesting, providing, or publishing services. Its main goal is to achieve better integration between both MAS and Service Technologies to support the development of open systems. For this same goal, the KAoS approach [UBL+08] has proposed the use of ontologies for a semantic representation of norms; i.e., *policies* according to the terminology defined by the KAoS proposal (described in Table 2.2, fifth row). One of the most interesting aspects of this language is the possibility of representing negative obligations, which are constraints that refrain agents from acting. Therefore, obligations specify actions that are required to perform, whereas negative obligations define actions for which such a requirement is waived.

Finally there are other proposals that do not represent deontic operators explicitly. The language described in [GC07] represents norms as ECA-rules (Event-Condition-Action) [DGG95], which employ the notions of ignoring, forcing, and expecting events, and preventing states (see the last row of Table 2.2). Therefore, it allows a more meaningful definition of norms. These rules define whether a forbidden action will be prevented from happening or whether it will be sanctioned if it occurs. In a more recent work [DTM09], Dastani et al. propose a language for programming NMAS. In this proposal, the notion of norms as deontic prescriptions does not appear explicitly. In contrast, they are defined by means of constitutive rules that define that some states of affairs count-as violations in the NMAS. In addition, this programming language is provided with an operational semantics that allow norms to be used explicitly as programming constructs.

Table 2.2 illustrates the main features that can be represented with each language. The first column details the *Deontic Modality* feature, which describes the kind of normative propositions that can be represented; i.e., permissions (P), obligations (O) and prohibitions (F). The *Control* feature determines whether the language defines constraints over states (*States*), actions (*Actions*), or both. In particular, some languages allow constraints over a specific set of actions such as illocutive or service access actions. In the case of the ECA-rule proposal, actions are the addition and removal of atomic formulae. The *Enforcement Mechanism* property considers whether it is possible to define sanctions and rewards for enforcing norms. The *Conditional* attribute determines the type of conditions (i.e., action or state condition) for activating the norm. And, finally, the *Temporal* feature represents if the language supports the definition of temporal constraints for norm activation such as "before", "after" and "between".

	Deontic Modality	Controls	Enforcement Mechanisms	Conditional Expressions	Temporal Constraints
Vazquez-Salceda et al. [VSAD04]	F/O/P	Action/ State	-	State	Before After
Electronic Institutions [GCNRA05]	F/O/P	Dialogical actions	Sanctions	Action/ State	Before/After Between
Contract [OLMN08]	O/P	State	-	State	-
Argente et al. [ACBJ08]	F/O/P	Service actions	Sanctions Rewards	Action/ State	Before/After Between
KaOS [UBL+08]	F/{-,+}O/P	Actions	-	Action	-
ECA-Rules [GC07]	-	Addition/Removal atomic formulae	Sanctions	State	-
Dastani et al. [DTM09]	- (Count-as)	State	Sanction Rules	State	-

Table 2.2: Comparison among languages for specifying norms

2.5.2 Operational Norms

The codification of norms by means of these normative languages is too abstract to be implemented in a society. Thus, norms must be interpreted or translated into *operational* norms that are meaningful for the society [GAD07]. In [VSAD04], Vázquez-Salceda et al. study the *operational* aspects of norms. These aspects, which are related to the development of agent platforms, should be taken into consideration in order to facilitate the implementation of norms.

The implementation of the *norm control* process consists of three different activities: (i) detection of norm activation; (ii) violation detection; and (iii) violation management. The norm control process is affected by the norm components, which are the following: (i) the norm target, which is the agent or agents affected by the norm; (ii) the controlled situation, which can be defined over a state or an action; (iii) the activation conditions; and (iv) the temporal constraints. A characterization of norms based on these components is shown below.

- *Norm Addressee.* The definition of control mechanisms should take into consideration the features of the norm addressee. According to these features, agents affected by norms are classified into:
  - External Agents: this group is formed by all agents that have been designed independently of the system. Thus, they should be highly controlled. Their mental states are not accessible, and their behaviour can only be observed by their public messages and visible actions.
  - Internal Agents: this set of agents consists of all agents whose goals and behaviours are known. Thus, control mechanisms for these agents are less important since the system designer has control over them.
- *Situation of the Norm.* In order to determine whether a norm has been violated, the controlled situation must be detected. As mentioned in the normative language description, the situation controlled by a norm can be defined over a state condition or an action. In general, the determination of a proposition is undecidable. In particular, it is true for complex logics, but in many implemented cases this is not the case. However, an in-depth analysis of decidability of propositions is beyond the scope of this chapter. On the other hand, the occurrence of public actions can be detectable. For norms controlling an action:
  - Obligation norms without temporal or activation conditions do not make any sense,

since they imply that the addressee agent is obliged to carry out the action continuously.

  - Unconditional permissive norms do not need control mechanisms. The occurrence of a permitted action or state of affairs does not imply a norm violation or fulfilment.
  - Finally, the control of prohibition norms consists of detecting the occurrence of the action. It can be implemented by means of a black list of actions and a trigger mechanism.

- *Conditional Norms.* In the case of conditional norms, both the condition that activates the norm and the norm deactivation (i.e., when the situation is satisfied, the activation condition is false or the expiration condition holds) must be detected.
  - The control mechanisms of obligation norms depend on the verifiability of their activation condition and situation. The control mechanisms consist of detecting the norm activation (i.e., the norm condition holds) and then the occurrence of the situation must be observed.
  - In the case of permissive and prohibitive norms, the occurrence of the situation controlled by the norm is first detected and then conditions are checked.

- *Temporal Norms.* This type of norm employs temporal expressions (i.e., after or before) in its definition.
  - Temporal permissive and prohibitive norms are controlled in a similar way to conditional norms; the situation is first detected and then the temporal constraints are checked.
  - The implementation of obligations with deadlines is more difficult, since deadlines must be checked before detecting the occurrence of the situation controlled by the obligation. In [VSAD04], Vázquez-Salceda et al. propose the employment of clock triggers for their implementation.

With regard to *norm verifiability*, the detection of norm violations depends on checking their verifiability, i.e., the possibility of carrying out the necessary checking. Thus, agents' actions must be observed and recognised as complying with or violating norms. According to the verifiability characteristics of norms, they are classified into:

- *Computationally verifiable norms*, in which conditions and controlled situations can be checked at any moment without any extra mechanism. Thus, these norms can be *monitored*, i.e., their conditions are observable and recognised as complying with or violating norms [MFM+09]. For example, a norm that claims that:

  > "any agent a belonging to the EI e is forbidden to utter illocution i"

  is computationally verifiable by the EI itself, since it has full knowledge about both the institutional state and the illocutions uttered by agents. Therefore, the institution itself is able to detect violations of this norm.

- *Non-computationally verifiable norms.* These norms are classified into: i) Norms that require extra resources for their verification. Since they cannot be checked at every moment, their verification is carried out at a specific moment (e.g., periodically or randomly) by arbiters [DDM02]. ii) Non-verifiable norms that have undecidable conditions or situations that cannot be observed. Consequently, the checking of this kind of norm cannot be carried out. For example, a norm that claims that:

  > "any agent a belonging to the EI e is forbidden to carry out action a which takes place out of the EI"

  is non-computationally verifiable by the EI itself since it has no observability of actions performed by agents outside the institutional boundaries. Some works on the implementation of non-computationally verifiable norms are described in Section 2.5.3.2 next.

2.5.3 Implementation Mechanisms

This last section focuses on the *implementation mechanisms* that are required to let norms have an effective influence on agent behaviours [GAD07]. These implementation mechanisms are classified into two categories (see Figure 2.1) [FC08, GAD07]: (i) *regimentation* mechanisms, which consist in making the violation of norms impossible; and (ii) *enforcement* mechanisms, which are applied after the detection of norm violations, reacting upon them. In a recent work [Bal09], a taxonomy of different techniques for effectively implementing norms was proposed. On the one hand, the regimentation of norms can be achieved by two processes: (i) *mediation*, in which both the resources and the communication channel are accessed through a reliable entity that controls agent behaviours and prevents agents from deviating from ideal behaviour;

and (ii) *hard-wiring*, assuming that the agents' mental states are accessible and can be modified in accordance with norms. On the other hand, norm enforcement techniques are classified according to both the observer entity and the enforcer entity. Norms are *self-enforced* when agents observe their own behaviour and sanction themselves. If those agents involved in a transaction are responsible for detecting norm compliance (i.e., *second-party* observability), norms can be enforced by: (i) the *second-party*, which applies sanctions and rewards as *retaliation* (i.e., to do something harmful to someone as a punishment), and *reciprocation* (i.e., to behave in the same way as someone else by helping each other and giving each other advantages), respectively; and (ii) a *third-party*, which is an authority and acts as an *arbiter* or *judge* in the dispute resolution process. In the case of *third-party* enforcement two other mechanisms for ensuring norm compliance can be defined according to the entity that is in charge of norm enforcing: (i) *social norms* are defended by the *society*; (ii) in *institutional enforcement* there are authorities in charge of monitoring and applying *institutional* sanctions and rewards.

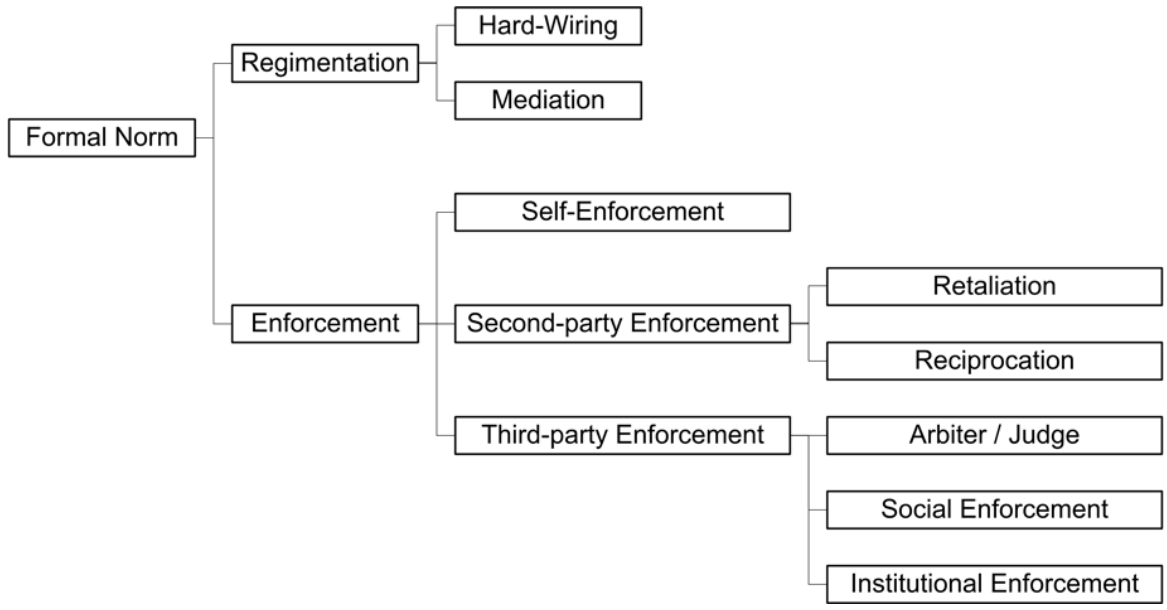

Figure 2.1: Operational interpretation of norms

2.5.3.1 Implementation of Norm Regimentation

Regimentation forces ideality (expressed as norms) and reality (defined by agents' behaviour) to coincide [JS93]. Proposals on the two main regimentation mechanisms are hard-wiring and mediation.

Hard-Wiring. The KAoS proposal, initially presented in [BDC+95], defines policies (i.e., message sequencing conventions) as a mechanism for coordinating agents' interactions (see Section 2.5.1). In this approach, policies governing conversational and other social behaviour among agents are defined off-line and are hard-wired in advance into agents. KAoS provides an explicit set of mechanisms for encoding message-sequencing conventions that, in most situations, release agents from the burden of elaborating inference that otherwise might be required to determine which messages are appropriate. Later, in [BUJ+03] the evolution of the KAoS framework is described. Specifically, in the new version of KAOS, policies are controlled using a mediation approach for controlling access to infrastructure and enforcement techniques for controlling obligations.

Mediation. In [GCRASV06], a formalism based on rules for representing constraints on agent behaviours is presented. This formalism is conceived as a "machine language" for implementing other higher level normative languages. The main features of the proposed "machine language" are: (i) it allows the explicit definition and management of agent norms (i.e., prohibitions, obligations and permissions); (ii) it is a general purpose language that is not aimed at supporting a specific normative language; (iii) it is declarative and has an execution mechanism.

This rule-based language has been used for enforcing norms that govern EIs [GCNRA05]. To implement this rule system, the Jess tool[5] has been employed as an inference engine.

In open systems, the regimentation of all actions can be difficult, but sometimes it is also inevitable or even preferable to allow agents to violate norms [Cas03]. The reasons behind desirability of norm violations are either that it is impossible to take a thorough control of all their actions; or agents could obtain higher personal benefits when norms are violated; or norms may be violated by functional or cooperative motivations. All these situations require norms to be controlled by enforcement mechanisms.

2.5.3.2 Implementation of Norm Enforcement.

This section reviews works on norm enforcement. These works have been classified according to the entity in charge of enforcing and monitoring norm compliance.

[5] http://herzberg.ca.sandia.gov/jess/

Self-Enforcement. This approach does not need the intervention of a third party. As far as we are aware, little work has been done on the definition of self-enforcement mechanisms in the MAS field. However, several proposals have considered the notion of self-enforcement from sociology [Vos01], economy [Mén00, TZ00], among others.

Second-party Enforcement. Second-party enforcement is characterised by the fact that those agents that are directly involved in an interaction are in charge of monitoring and taking coercive measures accordingly. In particular, *image* measures [MMH02, SS05](i.e., a subjective global or averaged evaluation of a target) is used to evaluate the behaviour of other agents. In [VS99], Venkatraman et al. propose an approach for testing the compliance of agents with respect to a commitment. Commitments are specified in temporal logic and their compliance is evaluated with respect to locally constructed models for the given observer.

Third-party Enforcement. This approach is characterized by the existence of a third entity in charge of applying sanctions in case of norm violations. However, these violations may be observed by: (i) an entity involved by the interaction in which the norm has been violated; and (ii) a third entity that observes an interaction in which it is not directly involved.

- **Second-party Observability**
  - *Arbitrating.* With this approach, norm fulfilment is controlled by a third entity (i.e., an *arbiter*) which undertakes a resolution process when a conflict among agents arises. This arbiter is informed about the contract or agreement established by some agents. Then agents involved with this contract inform the arbiter about their individual and subjective evaluation of the contract. The arbiter forms an opinion by analysing evidence and other sources of information such as reputation, others' observations, and so on and determines if remedial mechanisms should be applied. The work contained in [DDM02] proposes a framework for contract performance arbitrating. It uses subjective logic [Jøs01] as the formal basis for evidence-based reasoning. Subjective logic addresses the problem of forming a measurable belief about a proposition on the basis of insufficient evidence, or in the presence of uncertainty and ignorance.
- **Third-party Observability**

– *Social Enforcement.* In this case, there are agents that are not involved in an interaction that are capable of observing it. Thus they are capable of forming their own image about the interacting participants. Moreover, these evaluations can be exchanged, in a process known as *reputation* in the literature. In this case, the society acts as norm enforcer. Thus, agents are persuaded to obey norms since their non-normative behaviour can be observed by others who may refuse to interact with them in the future [SA07]. These non-compliant agents might even be excluded from the society [dPSS08]. The role of emotions in the enforcement of social norms is particularly interesting. There are works in social science that argue that the anticipation of emotions promotes the internalization and the enforcement of norms [Els96]. For example, the work described in [FvSM06] models the emotion-based enforcement of social norms in agent societies. In this approach, the society monitors norm compliance and generates social emotions such as contempt, or disgust in the case of norm violation, and admiration or gratefulness in the case of norm fulfilment. Similarly, agents observe the expression of these emotions and are able to generate emotions such as shame or satisfaction in response.

– *Institutional Enforcement.* In this mechanism, the institution itself is in charge of both observing and enforcing norms. Thus, in this approach there are infrastructural entities that act as norm observers and apply sanctions when a violation is detected. Distributed mechanisms for an institutional enforcement of norms are proposed in [GGCN$^+$07, MU00]. These works propose languages for expressing norms and software architectures for the distributed enforcement of these norms. The work described in [MU00] presents an enforcement mechanism that is implemented by the Moses toolkit [MU98]. Its performance is as general (i.e., it can implement all norms that are controllable by a centralised enforcement) and more scalable and efficient then centralized approaches. However, one of the main drawbacks of this proposal is the fact that norms can only be expressed in terms of messages sent or received by another agent; i.e., this framework does not support the definition of norms that affect an agent as a consequence of an action carried out independently by another agent. This problem is overcome by Gaertner et al. in [GGCN$^+$07]. In their approach, Gaertner et al. propose a distributed architecture for enforcing norms in EI. Specifically, dialogical actions performed by agents cause the propagation of norma-

tive propositions (i.e., obligations, permissions, and prohibitions). These normative propositions are taken into account by the normative level; i.e., a higher level in which norm reasoning and management processes are performed in a distributed manner. In a more recent work, Modgil et al. [MFM+09] propose an architecture for monitoring norm-governed systems. This work belongs to the Contract project. This architecture is formed by trusted observers that report to monitors on states of interest relevant to the activation, fulfilment, violation, and expiration of norms. This monitoring system is *corrective* in the sense that it detects norm violations and reacts to them. However, the *predictive* use of the monitoring system (i.e., to detect danger states) has been left as future work. In [ASP09], Artikis et al. have proposed a framework for the executable specification of NMAS, which is executed using the Causal Calculator[6]. This framework might be used for: i) *prediction* queries, which have an initial state and a narrative (i.e., a description of temporally-sorted externally observable events) as input and which computes the current social state (i.e., the permissions, obligations and sanctions that are associated with the members of the society); ii) *planning* queries, which generate norm compliant plans; and iii) *postdiction* queries, which determine the past states of that society.

2.5.3.3 Implementation of Mixed Approaches

Finally, there are works that use a mixed approach for controlling norms. They propose the usage of regimentation mechanisms to ensure compliance with norms that preserve the integrity of the application; however, enforcement is also used to control norms that cannot be regimented due to the fact that they are not verifiable or their violation may be desirable. An example on this mixed approach is shown in [CJBA10]. This work shows how those norms that define permissions and prohibitions related to the access to the organization are regimented through mediation, whereas obligation norms are enforced following the institutional sanction mechanism.

The ORA4MAS [HBKR10a] is another well-known proposal that makes use of a mixed approach for implementing norms. The ORA4MAS proposal defines *artifacts* as first-class entities to instrument the organisation to support agent activities within it. *Artifacts* are resources and tools that agents can create and use to perform their individual and social activities [ORV08].

[6] http://www.cs.utexas.edu/users/tag/cc/

Regarding the implementation of norms in the ORA4MAS framework, regimentation mechanisms are implemented in artifacts that agents use to access the organization (i.e., mediation). The enforcement of norms has been implemented by artifacts, which detect norm violations, and agents, which are informed about norm violations and carryout the evaluation and judgement of these situations.

The notion of artifact has not only been used by the ORA4MAS proposal. The programming language of Dastani et al. (described in Section 2.5.1) has also been used for the implementation of norm-based artifacts whose behaviour is specified in terms of regimented and enforced norms. Other proposals use artifacts for controlling norms, for example [AdBD10]. It proposes the definition of norms for timed agent systems. In addition, different strategies for implementing normative artifacts have been proposed according to the way in which norms are monitored and applied.

2.5.4 Open Issues for Implementing Normative Multi-Agent Systems

The great majority of the works on norm implementation are based on the existence of a shared reality which is fully observed by the institution. Thus, the institution is capable of both monitoring norm compliance and enforcing norms. However, this assumption of full observability is too strong in several domains. Nowadays, the evolution of Internet has brought about new open applications that are characterized by the interaction of heterogeneous and autonomous agents whose actions and activities may be under the control of different institutions. Norm implementation proposals must take into consideration the limitations that exist in open environments. The term "limitation" refers to the fact that an entity needs extra information and capabilities in order to act as norm supervisor or controller. Specifically, such limitations are related to the detection of conditions and the extra capabilities, such as the power to impose sanctions or rewards, which are needed o impose norms upon other agents.

Therefore, works on norm implementation must evolve in order to deal with the fact that there is not full observability of interactions among agents. The detection and reaction to norm violations should be carried out by institutional entities according to a partial observation of the real word. In this sense, future works might consider the problem of detecting norm violations on the basis of conflicts among agents. For these reasons, more effort in the development of

conflict resolution, arbitrating and judgement mechanisms for MAS is needed.

Finally, works on norm implementation should take a *predictive* perspective. The detection of potential non-compliant states and their avoidance at run time has not received enough attention in current proposals.

2.6 Norm Reasoning

The existence of a logic formalism for representing normative concepts allows them to be reasoned about. *Norm Reasoning* can be defined as the process of thinking about norms to make decisions [vdTT99a]. This problem can be tackled from two different perspectives [vdTT99a]: from a global point of view (*diagnosis systems*); or from an agent point of view (*decision making*).

The *diagnostic theory* is necessary at design time in order for the designer of the system to ensure that the system has adequate properties and for (the designer of) those agents whose interactions will be regulated to ensure that they conform to the system. In addition, diagnosis systems are also used at run time. From this perspective, they check systems against given principles or norms. In particular, diagnosis systems reason about incomplete past knowledge for distinguishing between norm violations and non-violations.

The *decision making* processes are necessary at run time because complex multi-agent systems usually need dynamic regulations. This is interesting, from the individual agent perspective, because norm adoption and compliance involve complex decision making. Decision making systems reason about the future actions of the agents and are guided by agent goals. In particular, they imply two different activities: making a decision about what goal to pursue and how this goal is going to be achieved. Thus, norms restrict the range of goals to be pursued and the set of actions available for achieving them.

2.6.1 Norm Decision Making Systems: Norm-Autonomous Agents

Since the development of norm-autonomous agents is the main focus of this thesis, in this section previous works that also have faced with this issue are described with more detail.

In [CCD99], Conte et al. mention that:

> In order to influence the behaviour of the agent, a norm itself must generate a corresponding intention; and in order to generate an intention it must be adopted

by the agent, and become one of its goals

Thus, a *norm-autonomous agent* is defined as an agent whose behaviour is influenced by norms that are *explicitly represented* inside its mind. It implies that *norm-autonomous agents* have capabilities for *acquiring* norms; i.e., agents are capable of recognising the norms that are in force their environment and managing normative beliefs. Moreover, agents may have some other motivations to *accept* a recognised norm and forming a normative belief. Besides that, agents are endowed with capabilities for determining whether a norm concerns their case and it is *relevant*. After the recognised norm has been accepted and considered as a relevant norm, then agents must make a decision to conform or not to it (i.e. forming a normative goal). This decision to execute a norm is called *norm compliance*. Also in [CCD99], Conte et al. also mention criteria for accepting or rejecting recognised norms, considering a given norm as relevant, and obeying or violating norms. For example, norms can be violated due to material impossibility to fulfil the norm, or since there is a conflict with a more priority goal. Therefore, this paper specifies what information should be considered for the recognition, acceptance and determination of norm compliance. However, this work does not describe how this information can be managed and considered by an agent. In this section, existing proposals on the development of norm-autonomous agents are described.

In [CDJT00], Castelfranchi et al. propose how an agent architecture can be extended with an explicit norm notion. Specifically, this architecture allows agents to represent norms explicitly and to deliberately follow or violate them. This work only proposes the architecture for norm-autonomous agents and details which tasks and deliberations should be carried out by agents. However, the authors do not specify how these tasks are performed. Thus, this proposal does not formalise which are the logical connections among norms and mental states.

Dignum et al. have proposed in [DMSC00] an extension of the classic BDI architecture for considering norms. The first issue addressed by this work is the explicit representation of norms that are used for inferring the agents' intentions. In addition, the classic BDI algorithm is modified with several steps that consider the existence of normative beliefs and the occurrence of new events related to the activation of norms. In this proposal, agents are capable of representing norms, determining when norms are active (i.e., relevant), and resolving conflicts among norms and existing intentions. However, agents do not have intrinsic motivations or goals. Their behaviour is completely determined by norms and no decision about norm compliance is carried out. Thus, agents follow blindly norms and conflicts are solved by means of

a static preference order among intentions.

The work of Boella & Lesmo in [BL01], is one of the first proposals on the MAS field that have provided a solution to the autonomous decision on norm compliance. According to this proposal, agents decide to comply with norms considering the consequences of obeying norms (i.e., the cost of norm fulfilment) and violating norms (i.e. the cost of sanctions). An important contribution of this work is that norm enforcers are considered as autonomous agents that have their own motivations and limited capabilities for detecting violations and applying sanctions. Thus, agents may have different motivations for violating norms: material impossibility, conflicts with other goals, the possibility of violating the norm without being detected, or the possibility of not being sanctioned. The decision about norm compliance is carried out by a utility function that takes care of all the above-mentioned factors. In this approach, agents have a static decision strategy and do not take into account on-line circumstances. Moreover, this paper does not provide any information about how the behaviour of the norm enforcer agent is modelled and how this information is used in the definition of the utility function.

These first proposals have made an important contribution by pointing out the main requirements for norm-autonomous agents. In addition, these early works provide intuitive ideas and recommendations to meet these requirements. For example, some strategies for making a decision about norm compliance are provided. However, enough details about how agent programmers can develop norm-autonomous agents that implement these strategies are not provided. More recent works have also confronted with the development of norm-autonomous agents. The agent architectures proposed by these later works can be classified into norm-oriented or goal-oriented according to the priority that agents give to norms with respect to their internal goals.

2.6.1.1 Norm-Oriented Agents

Norm-oriented agents have as main purpose the fulfilment of norms above the achievement of their internal goals.

NoA. In [Kol05], Kollinbaum has presented the noA proposal. It is a practical agent architecture with an explicit notion of obligation and prohibition. In this proposal, obligations are the agents' motivations, whereas prohibitions restrict the choices of activities that agents can

ideally employ. Basically, noA agents are aware of the activation (i.e., instantiation) and expiration of norms and determine which norms are relevant to the agent at a given moment. These instantiated norms are considered to select which plan will be executed. Thus, noA agents are norm-oriented agents that do not have internal motivations. Therefore, they will always try to fulfil all norms. Norm conflicts are the main cause of norm violation. Thus, this work does not consider the autonomous decision about norm compliance. In contrast, Kollinbaum's work is focused on the definition of algorithms and procedures for detecting and resolving norm conflicts.

Normative KGP Agents. Another example of norm-oriented agent is Normative KGP agents, which is described in [SST06]. This proposal consists in extending KGP *(Knowledge-Goal-Plan)* agents [KMS+04] with explicit normative notions such as obligations, prohibitions, and roles. Thus, norms define which are the responsibilities of a specific set of agents which are playing a given role. Therefore, agents consider as relevant all norms that affect the roles being played by them. KGP agents have internal motivations or goals. However, in case of a conflict between norms and goals, agents will always follow the behaviour specified by norms. Priority functions are used for solving possible conflicts among beliefs, goals, intentions, and norms. However, this work has not proposed any conflict resolution mechanism for making decisions about obeying conflicting norms. Therefore, KGP are not autonomous to decide which norms the agent wants to comply with. This proposal is more concerned about the consideration of norms to plan and decide which action the agent will perform next.

Gaertner's Proposal. The architecture proposed by Gaertner in his thesis [Gae08] is also an example of norm-oriented agent in which all norms are blindly followed. Specifically, this proposal has extended the multi-context BDI [PSJ98] proposal to consider obligations and prohibitions. These norms are translated into intentions. These new intentions might be in conflict with the previous ones. As a solution to this problem, Gaertner proposes the use of an argumentation-based approach and a preference function. However, this conflict resolution strategy does not consider norms explicitly; i.e., norms are translated into intentions. As being pointed out by Gaertner in his thesis, the addition of the explicit notion of norm in the multi-context BDI would add more flexibility and complexity to the normative reasoning; e.g., agents would de capable of reasoning about norm compliance. In addition, it would allow normative knowledge to be more easily distinguished; e.g., the explicit representation of norms allows

agents to evaluate other's behaviour with respect to norms. Moreover, Gaertner also claims that the graded version of the multi-context BDI architecture, which is proposed by Casali in [CGS11], will allow the development of normative agents more formally and on a much finer level of granularity.

2.6.1.2 Goal-Oriented Agents

In contrast to norm-oriented agents, goal-oriented normative agents have the main purpose of achieving their desires while trying to fulfil norms. Thus they have the capability of deciding about norm compliance.

BOID. In [BDH+01], Broersen et al. propose the extension of the BDI architecture with an explicit notion of obligation. This is one of the first proposals on norm-autonomous agents that describes how these agents (known as BOID) can be designed in practise. Thus, BOID agents are formed by four *components* that are associated with Beliefs, Obligations, Intentions and Desires. Obligations are the external motivations of agents and their validity is taken for granted. In this proposal, agents can violate norms only due to a conflict among obligations, desires or intentions. This type of conflicts is solved by means of a static ordering function that resolves conflicts between components and within components. According to the definition of these ordering functions, different types of agents can be defined. For example, agents in which the overruling order is B-O-I-D (i.e., beliefs over obligations, obligations over intentions and intentions over desires and intentions) give more priority to obligations that their internal motivations (desires) and blindly obey norms without considering their intentions. Agents can be goal-oriented or norm-oriented according to the definition of the ordering function. Therefore, agents always consider norms in the same manner; i.e., they cannot decide to follow or violate a given norm according to their circumstances. Thus, agents do not take a decision about norm compliance. In contrast, agents will give (or not) more priority to obligations than their internal motivations in a static and predefined way. This solution is suitable for controlled environments in which agents confront with foreseeable situations. However, complex scenarios in which agents should dynamically adapt require more flexible solutions to the norm compliance dilemma. As argued in [CDJT00] "if protocols that agents use to react to the environment are fixed, they have no ways to respond to unpredictable changes".

López y López's Proposal. One of the first proposals on goal-oriented agents that have explicitly considered the norm compliance dilemma is López y López's thesis [LyLLd06]. In this work, López y López has proposed both a model of norms for NMAS and an agent architecture for developing norm-autonomous agents capable of interacting in these norm-governed environments. Thus, agents in a NMAS are controlled by a set of norms that define the ideal behaviour. Therefore, the behaviour of any agent is influenced by the norms that are addressed to the roles that it is playing in a given moment (i.e., relevance). Moreover, agents are autonomous for accepting norms. Thus, agents must recognise the norm issuer as an authority. Besides that, any agent accepts a given norm since it has some reason to do it; e.g., the norm benefits to other agent that it wants to benefit. In addition, agents are autonomous for pursuing their own goals even if these goals violate the norms; i.e., agents are autonomous to come to a decision on norm compliance. Specifically, this work includes the notion of sanctions and rewards to persuade agents to follow the norms. In this work, López y López has developed different strategies to allow agents to make decisions about norm compliance assuming that there is a material system of sanctions and rewards. Specifically, these strategies are: *social*, which gives more importance to social goals than individual ones; *pressure*, norms with harmful sanctions are obeyed; *opportunistic*, only norms that are beneficial to the agent are respected; *fear*, all norms with sanctions are observed; *greedy*, norms whose fulfilment is rewarded are followed; and *rebellious*, no norm is respected. This work represents an important step towards the development of norm-autonomous agents capable of making flexible decisions about norm compliance. However, as pointed out by López y López in [LyLLd06], the deliberation about norm compliance is only based on the existence of an external mechanism of norm enforcement. Therefore, in absence of information about the enforcement mechanisms agents have no motivation to comply with norms. This proposal does not explain how agents comply with norms regardless of the existence of an enforcement system. However, there may be other motivations for norm compliance beyond the enforcement mechanisms.

EMIL. In all of the above-described proposals, norms are off-line programmed on agents or agents are on-line informed by authorities about norms. Therefore, agents are not capable of learning new norms on-line and adapting their behaviours according to these unforeseen norms. In relation with this feature, the EMIL proposal [ACCP07] has developed a framework for autonomous norm recognition. Thus, agents would be able to acquire new norms by observing

the behaviour of other agents that are located in their environments. Moreover, EMIL agents are also capable of determining the pertinence of a norm and its degree of activation; i.e., the norm relevance. Regarding norm acceptance, authors claim that EMIL agents accept norms unless there are good reasons not to do so. However, details about what are these good reasons are not provided and it seems that EMIL agents accept all recognised norms. Similarly, EMIL agents obey all recognised norms blindly without considering their own motivations. In a later work [AVC10], the EMIL proposal has been extended for allowing agents to make decisions about norm compliance and to internalize norms. The decision about norm compliance is made considering the expected utility that agents should obtain if they fulfil or violate the norm. As previously argued, the use of static decision-making procedures as utility or preference functions is unsuitable for developing agents that interact inside dynamic environments. In these unforeseen environments, agents' goals may change or even be unattainable and the utility functions may loose their validity.

Joseph's Proposal. In, [JcSSD10] Joseph proposes an agent architecture that allows agents to reason about norm acceptance. Specifically, norms are defined as unconditional obligations. Therefore, agents participate in argumentation processes for proposing, accepting, or rejecting obligations. In Joseph's proposal coherence has been used by agents as a criterion for rejecting or accepting norms that are proposed during the argumentation dialogues. Coherence theory studies associations; i.e., how pieces of information influence each other by imposing a positive or negative constraint over the rest of the information [Tha00]. Joseph's work is only focused on applying coherence to the norm acceptance dilemma. Thus, a norm is accepted or rejected considering the coherence of that norm with respect to the rest of the cognitive elements that re present in the agent theory. In contrast, norm compliance entails the understanding of the effects beyond norms. Therefore the autonomous decision about norm compliance requires a more complex notion of norm than the one used by Joseph. However, we consider that the coherence theory can be also a suitable solution for deciding about norm compliance in a more flexible way than priority or utility functions.

2.6.1.3 Norm Diagnosis Systems

Norm diagnosis systems can be employed by designers to check and verify properties of norms. They can also be used at run time in real applications for determining if norms have been

violated. This last usage of diagnosis systems is more related to the computational interpretation of norms. These works propose mechanisms for both detecting norm violations and for enforcing norms, they have been analysed in Section 2.5.3.

An important aspect of norms that regulate and coordinate MAS is their formal verification and analysis [Vas04]. On the one hand, the *verification* process consists of determining the coherency of the normative system (for a definition of coherence see Section 2.2.3). On the other hand, normative *analysis* consists of ensuring that the system has adequate properties according to the environment in which the norms will be applied. In other words, the analysis of norms implies checking the formal properties of norms. The verification can be seen as a special case of analysis in which only coherence properties are considered. Below, the main notable works on analysis and verification are briefly discussed.

In [RL07], Raimondi et al. present an algorithm for the analysis of NMAS. In this proposal, MAS are modelled as interpreted systems [Fag03], which are semantic structures that represent systems of agents. The notion of interpreted systems has been extended with deontic and epistemic operators to represent agent obligations and knowledge, respectively. In this work, the analysis of epistemic properties of MAS, and the correct behaviours are studied. The analysis is performed by model-checking. In addition, Raimondi et al. provide an implementation of the model checking algorithm together with experimental results. This work has been applied within the Contract project for the verification of systems based on contracts. Both the conformance of an individual contract participant to its contractually correct behaviour and the conformance of the combined behaviour of all the contract participants with respect to the overall contract can be verified with this approach.

Recent works on normative analysis take NTL logic [ÅvdHRA+07] as a basis for the definition of a normative system (which was previously explained in Section 2.4.5). One of the first properties of a normative system to be considered is its utility [ÅvdHW07]. Informally, the *utility* of a normative system can be defined as the difference between the utility of the system restricted by norms and the utility of the same system without applying any norm. As stated in Section 2.4.1, a normative system in NTL is defined as a set of transitions, which are labelled as legal or illegal. Therefore, the restriction of a system consists of removing all illegal transitions from the original system. The *utility* of a system, with respect to a goal priority hierarchy, can be defined in a simple way as the highest priority goal that is satisfied by the system. This notion of utility allows the definition of an *individually rational* normative system

to be one in which the utility of the normative system for each agent is higher than zero. This approach assumes that all agents respect the norms.

Other proposals have considered the analysis of normative systems located in an open environment, in which agents are able to decide to fulfil norms. As a consequence normative systems have been analysed taking works on Game Theory as a basis [BvdT06]. For example, in [ÅvdHW07] properties of normative systems are analysed from a game-theoretical approach. More specifically, the *strategic game* related to a normative system is defined as a formal game in which each agent has two strategies: to cooperate with the normative system and fulfil the norms or to violate all of the norms. Given this definition of a normative strategic game together with the previous definition of the utility of a normative system, a normative system is defined as *pareto efficient* if there is no normative system with higher utility. In the same way, a normative system is in *Nash equilibrium* if no agent can obtain a higher benefit in the event of its norm violation. A definition of the robustness of a normative system is proposed in a more recent paper [ÅvdHW08] . A normative system is defined as *robust* if it remains effective in the event of norm violation, i.e., the system can still achieve its goals even though an agent in the system deviates from ideal behaviour. A normative system, with respect to a set of design goals, is defined as *effective* when the application of these norms over the system allows it to reach its design goals. In addition, the computational complexity measures of the robustness property are also analysed.

The most relevant studies on norm verification are related to the detection and solution of normative conflicts. In [VKN07], a set of techniques for detecting and representing conflicts and inconsistencies among norms is proposed. A *normative conflict* arises when the same action is permitted and prohibited, whereas a *normative inconsistency* is defined as a situation in which the same action is defined as obliged and prohibited. Another interesting approach for solving norm conflicts by means of negotiation techniques [KNPS06] conceives norms as contracts between agents that cooperate. Agents are guided by different motivations, so conflicts among norms may then arise as a consequence. This work classifies different types of normative conflicts and proposes several methods for achieving an agreement among agents, which allows these conflicts to be overcome. This work also assumes that inconsistent norms are a prohibition and an obligation related to an action that has been defined as an atomic first-order logic formula. The influence area of a norm is defined as the set of actions affected by the norm. Thus, norm inconsistencies are classified into three main categories: (i) *inconsistent*, in which

all of the possible instantiations of the obligation are contained in the prohibition influence area and, consequently, there is no valid instantiation of the obligation; (ii) *partially consistent*, in which the influence areas are overlapped; and (iii) *consistent*, in which there is no conflict among norms. This work proposes the use of negotiation techniques for changing inconsistent situations into consistent ones. Finally, in [OLMN08], Oren et al. propose the use of heuristics that have been defined inside Argumentation Theory to solve normative conflicts. This work represents conflicts among norms as a graph in which the nodes are norms, and the arcs represent conflicts between norms. Basically, this work proposes the use of different heuristics to prune the graph and determine a set of non-conflictive norms.

2.6.2 Open Issues for Normative Reasoning

Open issues related to decision systems and diagnosis systems are explained in this section.

Although several proposals have been made to define *autonomous normative agents* [Cas99] endowed with capabilities for recognizing, representing, and accepting norms, and for solving possible conflicts among them, the definition of an agent architecture that overcomes all these challenges remains unsolved. As being illustrated by this section, in spite of the great amount of work that has been done to define norm-autonomous agents, the decision about norm compliance has not been addressed properly. Table 2.3 compares the main proposals on norm-autonomous agents described in this section. Specifically, this table illustrates performance of the proposed norm-autonomous agents with respect to their capabilities for: *representing* norms, *acquiring norms*, reasoning about norm *acceptance*, determining if a norm is *relevant* and reasoning about norm execution. The decision on executing a norm implies that this normative goal has been selected between internal goals and other normative goals. Thus, the decision on norm compliance subsumes the resolution of conflicts among mental propositions and norms. However, we would like to make a difference between those works that consider conflicts as the only cause for norm violations and those ones that consider the fact that norms can be deliberately violated in the absence of a conflict with another mental attitude. Therefore, Table 2.3 has two different columns, labelled as *norm compliance* and *conflict resolution*, in order to point out how the norm compliance dilemma is considered.

This table makes a summary of the proposals reviewed by this section. However, works of Conte et al. in [CCD99] and Castelfranchi et al. in [CDJT00] have not been compared in this table since they are more intuitive than formal. Specifically, these works specify which are the

	Norm Representation	Norm Acquisition	Norm Acceptance	Norm Relevance	Conflict Resolution	Norm Compliance
Dignum [DMSC00]	√			√	√	
Boella & Lesmo [BL01]	√					√
NoA [Kol05]	√			√	√	
Normative KGP [SST06]	√				√	
Gaertner [Gae08]					√	
BOID [BDH+01]	√				√	
López y López [LyLLd06]	√		√	√		√
EMIL [AVC10]	√	√		√		√
Joseph [JcSSD10]	√		√			

Table 2.3: Summary of proposals on norm-autonomous agents

main requirements for norm-autonomous agents but they do not propose any specific solution to meet these requirements. The first works on the definition of norm-autonomous agents have pointed out the role of norms on decision making and have explained from an abstract perspective the deliberation processes carried out by these agents. For example, Boella & Lesmo' in [BL01] have faced with the norm compliance problem by proposing the definition of static utility function that consider the cost of obeying a given norm and the possibility of being sanctioned. However, enough details concerning how this utility function can be defined are not provided. Later works, have tried to close the gap between these intuitive ideas and more specific frameworks that allow the practical implementation of agents built upon these ideas. Some of these later works have omitted the agents' autonomy and are focused on developing norm-oriented agents. Thus, they confront the problem of resolving conflicts among norms and other mental attitudes. In contrast, there are works that have faced with the agents' autonomy by using static mechanisms; e.g., the BOID architecture [BDH$^+$01] defines a priority order among mental attitudes. Therefore, agents will always consider norms in the same way. Thus, these goal-oriented agents always consider their goals as more important than any norm, independently of their circumstances. These static mechanisms entail a limitation on the agent capacities for adapting to new societies or to the environmental changes. The work of López y López, have explicitly proposed mechanisms for allowing agents to make a decision about obeying or violating a given norm at a specific moment. As being argued by López y López, in her proposal compliance with norms is only sustained by a material system of sanctions and rewards. Obviously, sanctions and rewards are one of the main motivations of agents when deciding to follow a norm. However, there are norms whose compliance is neither sanctioned nor rewarded. Moreover, decisions on norm compliance are expected to be more robust if norms are not only conducted by external sanctions [AVC10]. The present thesis represents and step towards the definition of flexible and complex decision mechanisms for norm compliance. In light of the promising results achieved by the coherence theory in the acceptance of norms, the work described here also proposes the use coherence as well as other factors (e.g., emotions) for making decisions about norm compliance.

Finally, a diagnosis system must provide mechanisms for detecting inconsistencies and redundancies among norms. Future works should consider the fact that norms may evolve and change over time, so these mechanisms must be applied at run time in order to resolve dynamic conflicts and inconsistencies. This issue is directly connected with the norm change problem

described in Section 2.7.

2.7 Norm Creation Process

Traditionally, two different approximations have been considered for establishing norms in agent societies [SC09]:

- *Top-down approach*, where the system designer defines the normative system statically off-line; or norms are created dynamically on-line by some agent that acts as a leader or a norm recommender. This second proposal is more suitable for open systems, in which structural, functional, and environmental changes might occur. Therefore, dynamic situations may cause the norms that regulate an organization to lose their validity or to be adapted over time. This approach is related to the *legalistic* perspective, described in Section 2.3.2.

- *Bottom-up approach*, which analyses how norms can emerge inside a group of agents. A norm has emerged when it is followed by a considerable portion of the society without being previously created. Therefore, cognitive autonomous agents might be able to create private norms based on their observations. This alternative fits with the *interactionistic* perspective, described in Section 2.3.2. Therefore, this approach is more suitable for the generation of methods, tools and techniques for NMAS aimed at controlling virtual communities that are characterized by the interaction between humans and agents.

2.7.1 Top-Down Approach

In this approach, also known as prescriptive, there is an institutional level that specifies how agents should behave. In particular, norms are created off-line by the system designer or they are created on-line by a *legislative* agent empowered to change the normative system.

2.7.1.1 Off-Line Creation

Norms are created by the system designer and are usually regimented by hard-wiring or mediation. In the former mechanism of regimentation, agents are built according to norms and they cannot deviate from the desired behaviour. One of the most representative works belonging to

this category is the *social law* approach (see Section 2.4.4). A well-known example of mediation are the Electronic Institutions (EIs), where the system designer specifies all the elements of the EI (i.e., the dialogic framework, performative structures, scenes, ontologies, illocutions, and norms) before the institution is executed [EdlCS02]. As previously argued, the off-line approach is more appropriate for closed and homogeneous systems in which all agents have been created by the system designer and norms are always fulfilled.

2.7.1.2 On-Line Creation

Norms may be changed on-line by agents to adapt to changing environments. The use of *legislative norms* to create, modify, or abolish norms from the system is proposed in [LyLL02] (Section 2.4.4). Therefore, legislative norms define when and who is authorized to carry out legislative actions, which comprise at least three functions, namely issuance, abolition, and modification of norms.

Works on the dynamic creation of norms by legislative agents are becoming more and more important. One of the most recent works on the on-line creation of norms is presented in [CR09] and later extended in [CRP10]. These proposals consist in using planning techniques for synthesizing norms. Specifically, prohibitions are created to avoid undesirable states but to allow agents to reach their own goals.

The *norm change* topic consists in analysing the type of dynamics involved in systems of norms. The formalization of the norm change process has inspired the main issue covered by the workshop of *Formal Models of Norm Change (I and II)*[7]. In this international workshop, norm change has become the main topic; therefore the papers presented here range from diverse topics such as logic, game theory, and agent based approaches to norm change. Other works on the formalization of norm change include Boella et al., who presented an abstract model for norm change in [BPvdT09a]. They define normative systems as sets of input/output (see Section 2.4.2) rules (i.e., norms). Normative systems can be modified by means of contraction and revision operators. In this work, the authors make a deep analysis of norm change operators with respect to belief changes [AGM85]. In a more recent work, Boella et al. [BGRvdT10] propose a mechanism for adapting norms to unforeseen situations. Specifically, they propose to modify the conditions that define the applicability of regulative norms when these norms do not achieve their purpose. In [TDM10], the problem of norm change has been addressed

[7] http://www.cs.uu.nl/events/normchange2/

from a more practical perspective. Specifically, this work proposes a computational language for programming the run time modification of abstract norms and the concrete instantiations of norms (see the *Contract* explanation in Section 2.5.1 for a description of the normative language used by this proposal).

According to these approaches, norms are created by a leader or an agent that is endowed with the capabilities for it; however, norms are followed or adopted for different reasons such as: fear of sanctions [LyLLd02], leadership [Ver00], among others. Simulation proposals on the adoption of norms designed on-line are briefly described below.

Sanction Mechanism. These models include the notion of sanctions to punish agents that do not follow the norms. The work proposed in [LyLLd02] consists in experimentally comparing different strategies for norm adoption, given that there is a material system of sanctions and rewards. These strategies are: *social*, which gives more importance to social goals than individual ones; *pressure*, norms with harmful sanctions are obeyed; *opportunistic*, only norms that are beneficial to the agent are respected; *fear*, all norms with sanctions are observed; *greedy*, norms whose fulfilment is rewarded are followed; and *rebellious*, no norm is respected. In order to determine how norms influence agents, the *individual satisfaction* of an agent is measured as the percentage of the agent goals that have been satisfied. Similarly, the *global satisfaction* has been calculated in order to analyse the influence of norms on the society. According to the experimental results, social strategies make societies more stable, since social goals, which are expressed by norms, are almost always guaranteed [LyLLd02]. Moreover, in those scenarios in which agents give a higher priority to their own goals, there is no guarantee that higher levels of individual satisfaction will be achieved. In this case, the existence of conflicting goals as well as the application of punishments are obstacles to the satisfaction of individual goals.

Leadership Mechanism. The social structure is a key aspect in the creation of norms, since the network provides the infrastructure for the norm exchange. In addition, as occurs in human societies, agents may belong to groups or associations. In these groups, there are members that act as leaders and influence the group. Regarding this notion of leader, Verhagen makes an analysis of how the acceptance of norms changes agent behaviour in [Ver00]. More specifically, this work is focused on the norm spreading and internalization processes. In this work, Verhagen defines the notion of norm advice, i.e., there are agents that recommend norms acting as *leaders* of the society. Following with the notion of leadership, a role-based model

for creating norms is proposed in [SCPP07]. In this model, *leaders* are normative references for other agents that request advices about the adoption of norms. However, agents maintain their autonomy for deciding whether to follow a norm. This work also shows the performance of this role-based model with respect to the topology of the network in which the recommended norms are distributed.

2.7.2 Bottom-Up Approach: Dynamic Emergence

According to the Merriam-Webster[8] dictionary, to emerge means "to come into being through evolution". This section focuses on the emergence of norms inside artificial societies that are populated by software agents.

Within the MAS community, the *emergence* of norms has been defined by Conte et al. in [CACP07] as a macro-level effect of interactions among agents, which are carried out at the micro-level (see Figure 2.2(a)). However, changes in norms at the macro-level also affect the micro-level, since agents learn and internalize norms inside their minds (this process is known as *immergence*). Therefore, norm dynamics can be represented as a cycle created by the emergence and immergence processes. In [CACP07] Conte et al. have made an in-depth analysis of how the macro level affects the micro level in different ways. Therefore, the *emergence* of norms is closely related to the process by which agents incorporate and internalize norms (for a description of works on normative reasoning see Section 2.6.1). Along the same line of thought, in [BvdT04b], Boella et al. also argue that the process by which norms emerge is a cyclic process, called as the *Social Delegation Cycle* (see Figure 2.2(b)). This cycle explains how norms emerge from agents' desires in three steps. Specifically, group goals are built upon agents' desires. Then, group goals are transformed into social norms (emergence). Finally, these norms are accepted and internalized by agents (immergence). Despite the fact that the two cycles are quite similar, the approach and focus of these works is not the same. The proposal of Conte et al. is focused on how normative agents recognise and internalize norms. Thus, this work is close to norm immergence. With this aim, they have proposed an agent architecture that has been described in Section 2.6.1. In contrast, the work of Boella et al. deals with the emergence of norms from a theoretical point of view. More precisely, their main contribution consists in a formalization of the norm dynamics based on their model of NMAS (which has been described in Sections 2.3.1.1 and 2.3.2) and their proposal on Input/Output logic (also commented in Section 2.4.2).

[8] http://www.merriam-webster.com/

The authors formalize the definition of group goals as an aggregation of agents' desires, whereas norms are created by a planning algorithm that considers obligations, sanctions, and rewards. Finally, the norm acceptance has been formalized taking the perspective of game theory.

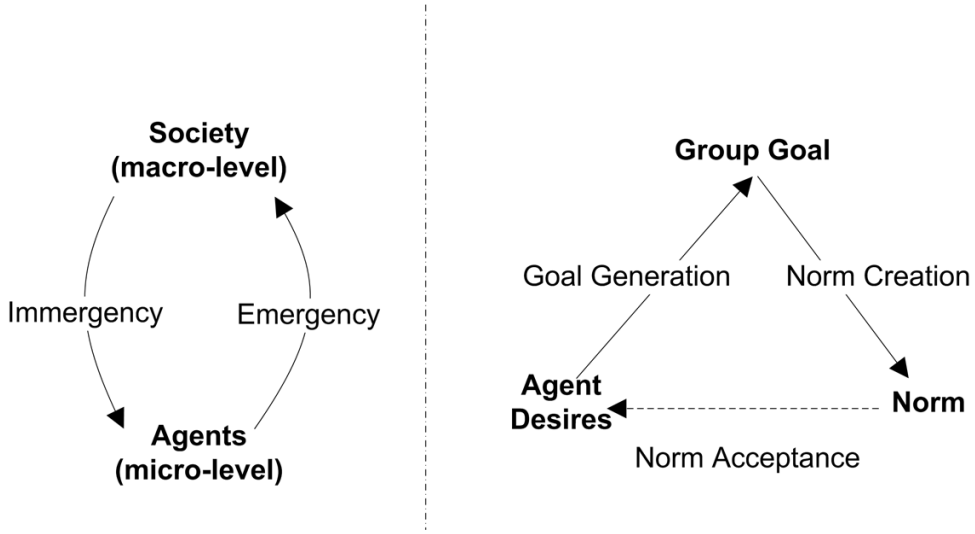

Figure 2.2: (a)Emergence in the Loop [CACP07] (b)Delegation Cycle [BvdT04b]

The work contained in [SC09] reviews simulation works on the emergence of norms that are classified according to the norm emergence mechanisms. Some of the most relevant works on the emergence of norms are described below.

Imitation Mechanism. According to this mechanism, agents follow the behaviour exhibited by the majority of the members of the society. One of the first works in this direction was made in [Eps01]. In this model, agents learn how to behave (what norm to adopt) by imitating the most commonly followed pattern of behaviour. In order to determine the norm or pattern that is most followed, agents observe the behaviour of those agents belonging to their observation radius. The main limitation of the imitation approach is that only one norm can emerge (i.e., the most followed norm), so it cannot explain the co-existence of multiple norms [NL04].

Machine Learning Approach. One of the first approaches on the use of machine learning algorithms to norm emergence was presented in [ST92a] and later in [ST97]. In these works, Shoham and Tennenholtz use a reinforcement learning algorithm to allow agents to reach an agreement on conventions. Conventions (see Section 2.3.1.2) are simple norms that impose restrictions over agents' behaviours. Specifically, conventions define what particular game strategy is considered as allowed. This work has influenced later proposals. For example,

Walker and Wooldridge's work [WW95] propose several strategies for the definition of norms. The goal is to experimentally analyse different functions for reaching an agreement on adopting a norm. The experimental results show that strategies that quickly allow agents to reach an agreement imply agents are more willing to change their adopted norms. In a more recent work [Puj06], Shoham and Tennenholtz's algorithm has been used as a mechanism for the emergence of social conventions. In particular, Pujol's proposal consists in analysing the role of the network topology, in which agents interact, with respect to the emergence of conventions.

An interesting experiment on the emergence of social norms via private interactions is shown in [SA07]. The main goal of this work is to analyse whether norms can emerge inside open and heterogeneous societies where agents interact privately with multiple agents. This way of learning has been defined as *social learning*. In this work, Sen and Airiau study the effect of different social learning algorithms, population size, etc. on the convergence to a norm. The experimental results of this work confirm that social learning is a robust mechanism for reaching an agreement about adopting a norm. Finally, [SPP08] describes a set of experiments that study the effectiveness of social norms inside open environments where the number of agents to be controlled may be huge. Specifically, the results of this work demonstrate that social norms that are enforced by distributed peer-to-peer punishment are more suitable for those constraints whose enforcement cost is low. However, for those situations that are less frequent and require larger costs for punishment, institutional norms (i.e., laws) and sanctions are more appropriate.

In all of the above proposals, agents do not have an explicit consideration of norms. On the contrary, norms are strategies or behavioural rules. Proposals on norm emergence that consider norms as an expectation that predicts others' behaviour are described below.

Emotion-Based Approach. The work described in [FvSM06] considers the role of emotions in the norm emergence process. As previously mentioned, Fix et al. propose a model of norm implementation in which the society acts as norm enforcer by imposing sanctions and rewards by means of emotions. Any agent that perceives compliance with norms expresses social emotions such as contempt, disgust, admiration, gratefulness, among other emotions. These emotions can be perceived and used to recognise the social norms.

Cognitive Approach. This line of research is mainly represented by the EMIL-A proposal, which is described in [ACCC08]. In this work, agents are endowed with cognitive capabilities that allow them to infer norms from their observations. In particular, Andrighetto et al. analyse the role of agents' mental capabilities on the emergence of norms. They propose a mechanism that allows agents to learn norms governing their environments. The main differences of this approach with respect to imitation and machine learning are: (i) agents are not utilitarian (i.e., agents are not necessarily utility maximizing), as in the case of machine learning; (ii) agents are capable of recognising a set of norms, not a single norm as in imitation; and (iii) agents consider norms as expectations of behaviours in their minds.

2.7.3 Open Issues for the Emergence of Norms

A general problem of the works on the top-down and bottom-up approaches is that they do not make a distinction between recognising and obeying norms. Thus, learning a norm does not imply complying with it. Therefore, works should make a clear distinction between these two processes and make a deeper analysis on the relationships.

Next, specific open issues for top-down and bottom-up approaches are described:

- The top-down approach entails the development of legislative agents. Therefore, these agents can identify when the norms should be adapted and how to adapt them properly. These tasks are not only related to regulative norms (see the norm typology contained in Section 2.3.1.1). For example, they must also adapt the set of institutional concepts and decide when one concept has been subsumed by another. As a consequence, a study of mechanisms for the emergence of regulative and constitutive norms should be carried out.

- Works on the bottom-up approach should take their inspiration from real social applications (e.g., Second Life[9]). In this kind of scenario, norms are controlled by means of social recriminations. Thus, the MAS area must study these scenarios to analyse social dependencies among individuals (i.e., social relationships) and the way in which the society creates and enforces norms.

- Another issue which, in our opinion, must be considered more deeply is the role of emotions in the emergence of norms. The works described in this section use emotions as a

[9] http://www.secondlife.com

mechanism for identifying norms. The role of emotions in the norm compliance dilemma should be analysed to provide a more realistic and complex solution to the norm reasoning problem. Traditional proposals on reasoning about norm compliance deal with this problem from a utilitarian perspective. Thus, decisions are made by considering the effect of norm compliance on agent goals, but they ignore the emotional state of the decision maker. Finally, future works should analyse the role of emotions as heuristic information to determine when a decision about following or violating a norm should be reconsidered.

2.8 Conclusions

This chapter presents an overview of the most relevant works on norms for MAS. This review takes inherent problems in open systems as a basis and points out the main deficiencies and drawbacks of current proposals when supporting open MAS. The main conclusion of this review is that there are many questions that still pending. Next, some of these open issues and challenges for the development of NMAS are detailed.

2.8.1 Specification of Normative Systems

The complexity and dynamical features of open applications require a formal language to represent both the norms and the normative system. This issue can be divided into:

1. Definition of a normative model in order to design complex and dynamic agent societies. This model should: (i) cover the norm levels and give support to the different types of norms (i.e., constitutive and regulative); (ii) permit the dynamic adaptation of norms; and (iii) permit the definition of mechanisms to support the different levels of norms and take into consideration the relationships among them.

2. Development of a computational logic language for the specification of normative systems. This language should be expressive enough to allow the definition of complex systems. In addition, methods and techniques for automatically reasoning about normative systems must be developed. Thus, new tools for simplifying and checking the consistency of normative systems on-line are needed. This entails working on the analysis of normative systems such as model-checking, specification, verification and analysis, consistency, coherency, completeness, redundancy and simplification, and so forth. Time complexity

must be also taken into account, from a theoretical (i.e., complexity analysis of algorithms) and experimental (i.e., real execution times) perspective in order to allow NMAS to be used in more real-time applications. Finally, the possibility to distribute this reasoning among different entities is also a key factor for massive applications in which thousands of users are participating simultaneously.

3. The dynamical feature of open environments may require the adaptation of norms. Legislator entities require mechanisms and tools to detect when a norm is needed, when there are redundant norms, or when they lose their validity. Thus, the dynamism and unpredictability of open MAS entails the development of tools for detecting emergent patterns of cooperation and translating them into norms.

2.8.2 Individual Normative Reasoning

Sophisticated agent architectures and decision making procedures that allow agents to have explicit knowledge about norms and to be able to decide about convenience of norm compliance are also needed:

1. The set of norms that regulate a NMAS may dynamically evolve over time. Therefore, agents must be endowed with capabilities for recognising and acquiring new norms at run time and consider them in their decisions. These recognition mechanisms must consider different sources of information as clues for detecting norms. These sources are classified into: (i) explicit normative perceptions, which correspond to those messages exchanged by agents in which norms are explicitly communicated; and (ii) implicit normative perceptions, which include the observation of actions and emotions.

2. Moreover, decision-making procedures for making a choice about obeying or violating norms must be developed. Existing proposals consider the impact of norm compliance over agent goals (i.e., the expected utility of obeying norms). However, the rational decision about when a norm must be violated deliberately must be considered by future works.

3. Making a decision about violating (or complying with) a norm must consider the expected utility of this decision in terms of the effect on agent's goals, the coherence of this decision

with respect to the agent's cognitive elements, and the emotional consequences of these decisions.

4. Emotions have been proposed as a source of information for determining when an intention must be aborted. For example, the intensity of emotions such as fear determines when an intention must be abandoned, and it is necessary to search for alternatives to achieve a specific goal. Similarly, emotions might be used for determining when and how decisions concerning norm compliance must be reconsidered.

2.8.3 Implementation of Norms

The implementation of norms must consider the difficulties that arise in real scenarios. The majority of norm implementation mechanisms have been built assuming the existence of a shared reality that is fully observed. However, in real scenarios, agents interact within *uncertain* environments. There may be different reasons for uncertainty such as: agents have a *limited* and not fully believable knowledge of the world; there may be *ambiguous* interpretations causing doubts, conflicts, or confusion. These issues should be taken into account by future works on the implementation of norms:

1. The uncertain environment implies a drastic evolution of the violation determination process. Until now, norm violations have been detected by observing agent behaviour. Uncertainty about norm violation is explained by two main reasons: the opacity and limited knowledge about actions and illocutions performed by agents; and the existence of subjective conditions of norm violation due to the ambiguous interpretation of norms. Moreover, norm violations may also be caused since agents are either unaware of the existence of norms or do not perceive the discrepancy between the norm and their behaviour. As a consequence, in practice, norms are not logical formulas that define what is considered as obliged, permitted, and forbidden. Actually, norms involve processes to determine whether a violation has occurred according to what has been observed by agents.

2. Agents should be able not only to make a decision about norm compliance, but also to confront situations in which there is a conflict about the violation of norms. The norm-implementation mechanisms should make use of both conflict resolution and argumentation techniques for reaching a consensus about norm violation and determining responsibilities and redresses.

3. Conflict resolution processes imply endowing agents with capabilities for providing explanations for their actions. Action justification means providing an explanation to the norm compliance dilemma; i.e., to account for the reasons and circumstances in which the norm compliance dilemma has been solved. In addition, the fact that norms can be interpreted ambiguously implies that agents should justify how norms have been interpreted.

2.8.4 Software Tools for Normative Multi-Agent Systems

New software tools are needed in order to solve real problems that are modelled as NMAS applications. These tools must assist developers in:

1. Design of NMAS applications. These tools must provide the system developers with guidelines for modelling such complex systems. More specifically, they must facilitate the specification of agent societies as well as the norms that allow the agent activities to be controlled and coordinated.

2. Implementation of NMAS applications. Implementation tools must provide support to normative agents, but they must also provide mechanisms for adapting normative structures in response to environmental changes.

    - The infrastructure of the NMAS must provide agents with mechanisms for creating, communicating and spreading norms. It must allow the definition of new regulations, normative terms, and legal contracts.
    - To deal with undesirable behaviours, the NMAS must provide agents with enforcement and monitoring mechanisms.
    - To allow external agents, which may not necessarily be norm aware, to behave according to the normative system, tools for automatically processing norms are needed.

Open issues in NMAS aimed at supporting virtual communities have been described throughout this work. These new societies raise new complex challenges that must be approached from different disciplines such as ethics, law, and sociology. As illustrated by this review, the works proposed by the different researchers are quite heterogeneous even if they address similar problems. There is a lack of standardization that makes the comparison and combination among

the different proposals difficult. As a consequence, a higher degree of agreement within the NMAS field would help to overcome these open issues.

Chapter 3

Normative Definitions

In this chapter, we introduce the norm definitions that we use over the course of this thesis. Norms are coordination mechanisms that attempt to: (i) promote behaviours that are satisfactory to the organization, i.e., actions that contribute to the achievement of global goals; and (ii) avoid harmful actions, i.e., actions that prompt the system to be unsatisfactory or unstable. The *norm* concept is defined by the Encyclopaedia Britannica [1]as:

> "a rule or standard of behaviour shared by members of a social group. Norms may be internalized; i.e., incorporated within the individual so that there is conformity without external rewards or punishments, or they may be enforced by positive or negative sanctions from without. [...] Norms are more specific than values or ideals: honesty is a general value, but the rules defining what is honest behaviour in a particular situation are norms"

According to this definition, norms guide the behaviours of those ones that belong to a group; i.e., they are aimed at achieving coordination inside groups. They have been studied from different perspectives such as philosophy [vW63], sociology [Sea69], law [AB71], etc. MAS research has given different meanings to the norm concept. For example, it has been employed as a synonym of obligation and authorization [Dig99], social law [MT95], social commitment [Sin99] and other kinds of rules imposed by societies or authorities.

As being pointed out by several works on a philosophical study of norms [Raw55, Sea69, Sea97], systems of norms are formed by different types of norms. Specifically, there are two main types of norms: *constitutive* and *deontic*. On the one hand, deontic norms are guides for action

[1]norm. (2010). In Encyclopaedia Britannica. Retrieved November 17, 2010, from Encyclopaedia Britannica Online: http://www.britannica.com/EBchecked/topic/418203/norm

formulated on the basis of experience. They are expressed in terms of obligations, permissions and prohibitions, so they are named as deontic. On the other hand, constitutive norms define the institutions that are regulated through deontic norms. Thus, constitutive norms define how the institutional reality (i.e., the institutional facts) is built in terms of actions or state of affairs occurring in the real world (i.e., brute facts). The purpose of this thesis is not to propose, compare or improve these existing norm models. Therefore, this chapter contains the normative definitions that have been used in this thesis. Specifically, this chapter is structured as follows: Section 3.1 contains an introduction and some preliminary definitions; Section 3.2 contains our definition of deontic norm; Section 3.3 contains the definition of constitutive norm; and Section 3.4 contains the conclusion to this chapter.

3.1 Introduction

The normative model used in this thesis [CAB11c, CAB09b] was originally proposed with the aim of providing a model of norms for controlling Virtual Organizations (VOs). VOs are a cooperation of legally independent enterprises, institutions or individuals, which provide a service on the basis of a common understanding of business [MFPF01]. Therefore, the original model was focused on allowing norms to be used in MAS applications. However, this thesis is focused on reasoning about norms from the agent perspective. As a consequence the original model has been redefined in this chapter according to the purpose of this thesis.

Definition 3.1.1 (Normative Specification) *The Normative Specification (N) of a MAS is the set of norms that control the MAS. It is defined as:*

$$N = N_{Deontic} \cup N_{Constitutive}$$

where

- $N_{Deontic}$ *is the set of deontic norms that define what is considered as prohibited, forbidden or obliged.*

- $N_{Constitutive}$ *is the set of constitutive norms that define the anchoring among the real world and the institutional reality.*

In this model norms (N) are classified into two categories: *Deontic* ($N_{Deontic}$) and *Constitutive* ($N_{Constitutive}$) norms. *Deontic* norms merge the notion of *procedural* and *regulative* norms described in Section 2.3.1.1. Thus, they define a deontic control (i.e., a permission, prohibition or obligation) over an action or a state of affairs. In addition, they might define the enforcing mechanisms in terms of punishments and rewards carried out by representative agents of the MAS. Therefore, *deontic* norms define a practical connection between a regulation and its consequences [BvdT08]. *Constitutive* norms define how actions or state of affairs taking place in the real world (i.e., brute facts) modify facts on the institutional state (i.e., institutional facts).

In the following sections, we formalize our notions of deontic and constitutive norm. In these definitions, we take as a basis the formalization of norms made in [OLMN08]. In this proposal a distinction among *norms* and *instances* is made. A *norm* is a conditional rule that defines under which conditions it must be instantiated. *Instances* are created out of the *norms* when their activation conditions are satisfied. These norm instances remain active until their expiration conditions hold. Before defining deontic and constitutive norms and instances, we must provide some preliminary information.

Preliminaries. Let us suppose the existence of a first-order predicate language $\mathcal{L}$ whose alphabet includes: the logical connectives $\{\wedge, \vee, \neg\}$; parentheses, brackets, and other punctuation symbols; and an infinite set of variables. Variables are universally quantified implicitly. Along this thesis variables are written as any sequence of alphanumeric characters (including '_'[2]) beginning with either a capital letter or '_'. In addition, the alphabet contains non-logical predicate, constant and function symbols, which will be written as any sequence of alphanumeric characters beginning with a lower case. Specifically, there are constant symbols that identify roles, agents and agent institutions. Thus, $\mathcal{R}$, $\mathcal{A}$ and $\mathcal{I}$ are the sets containing all role, agent and institution identifiers, respectively. The set of predicate symbols is formed by *action* predicates ($\mathbb{X}$) and *state* predicates ($\mathbb{P}$), which describe properties of the world and the institution. Thus, the institutional predicates are subset of the predicates ($\mathbb{I} \subseteq \mathbb{P}$) describing the institutional state, whereas brute predicates ($\mathbb{B} = \mathbb{P} \setminus \mathbb{I}$) are those facts describing changes in the world produced by the actions of agents. Finally, given any set of predicates $\mathbb{A}$, $Lit(\mathbb{A})$ represents the set of atomic formulas built from their predicates and their negation. Let us also assume the standard definition for *wffs* (well-formed formulas). Thus, *wff* denotes a single

[2] '_' stands for an anonymous variable that matches anything.

well-formed formula. We will make use of the standard notion of substitution of variables in a *wff*; i.e., σ is a finite and possibly empty set of pairs Y/y where Y is a variable and y is a term.

3.2 Deontic Norms

Deontic norms define patterns of behaviours by means of *deontic modalities*: *obligations*, which define which actions or goals should be performed or satisfied by agents; *prohibitions*, which define which actions or goals should not be performed or achieved; and *permissions*, which define exceptions to the application of a more general norm of obligation or prohibition. Therefore, deontic norms define a pattern of behaviour (or *norm condition* in our terminology) as obligatory, prohibited or permitted. This norm condition can be represented as actions to be performed or goals to be achieved. In fact, we make no sharp distinction between actions and goals, since what in one situation is best described as an action may be best described in another situation as a goal [LyLLd06]. Also inspired by the representation of [OLMN08], we define deontic norms as conditional rules that are relevant to a set of agents under specific circumstances. Thus, the set of agents that is affected by a specific deontic norm are the ones that are playing the *target* role of this norm. In this way, deontic norms represent the responsibilities, rights and duties of roles with respect to the organizational goals. In general, norms are not applied all time, so they include the notions of activation and expiration conditions. Specifically, the *activation condition* defines when obligations, permissions and prohibitions must be instantiated and must be fulfilled by all agents playing the target role. Instances remain active, even if the activation condition ceases to hold. Specifically, the *expiration condition* defines the validity period or deadline of the norm instance. Finally, inspired by [LyLLd06], deontic norms also include information about the enforcement mechanisms: *sanctions*, to punish agents which do not obey the norm and *rewards*, for rewarding norm fulfilment. Both sanctions and rewards are the means for the target agents to know what might happen whatever decision they take regarding deontic norms.

3.2.1 Deontic Norm Definition

Given the informal definition of norm and the logic preliminaries given above, a deontic norm is formally defined as:

Definition 3.2.1 (Deontic Norm) *A deontic norm (n_d) is defined as a tuple* $n_d = \langle D, C, T, A,$

$E, S, R\rangle$, where:

- *$D \in \{\mathcal{O}, \mathcal{F}, \mathcal{P}\}$ is the deontic modality of the norm, determining if the norm is an obligation ($\mathcal{O}$), prohibition ($\mathcal{F}$) or permission ($\mathcal{P}$);*
- *C is a wff of $\mathcal{L}$ that represents the norm condition, i.e., it denotes the goal or action that is controlled by the deontic norm;*
- *$T \in \mathcal{R}$ is the target of the norm; i.e., the role to which the norm is addressed;*
- *A is a wff of $\mathcal{L}$ that describes the activation condition;*
- *E is a wff of $\mathcal{L}$ that describes the expiration condition;*
- *S is a wff of $\mathcal{L}$ that describes the sanction that will be applied to the target agents if the deontic norm is not fulfilled;*
- *R is a wff of $\mathcal{L}$ that describes the reward that will be provided to the target agents if the deontic norm is fulfilled.*

Let us suppose that there is a software agent, which will be named as *assistant*, that draws up traffic routes according to the preferences that a human user has specified. In order to calculate the most suitable route according to the user's preferences, that *assistant* agent needs to know which are the norms that regulate traffic in each region. For example, it is very usual that there are traffic laws that prevent accidents. For example, a deontic norm that obliges all car drivers to slow when there is heavy rain in some area (A) is represented as follows:

$$\begin{gathered}\langle \mathcal{O}, slow(A), carDriver, \\ heavyRain(A), \neg heavyRain(A), penalty, -\rangle\end{gathered} \qquad \text{(Heavy Rain Norm)}$$

Thus, the *assistant* agent knows that when the planned routes cross an area where there is heavy rain the speed must be reduced.

3.2.2 Deontic Instance Definition

Once the activation conditions of a deontic norm hold it becomes active and several instances, according to the possible groundings of the activation condition, must be created. Thus, *deontic instances* that are created out of the deontic norms are a set of unconditional expressions that

bind a particular agent (i.e., *the target agent*) to an obligation, permission or prohibition. Formally a deontic instance is defined as:

Definition 3.2.2 (Deontic Instance) *Given a deontic norm* $n_d = \langle D, C, T, A, E, S, R\rangle$ *and a theory* Γ *of* $\mathcal{L}$*, a deontic instance of* n_d *is the tuple* $i_d = \langle D, C', T', AgentID, A', E', S', R'\rangle$ *where:*

- $\Gamma \vdash \sigma(A)$ *where* σ *is a substitution of variables in* A *such that* $\sigma(A)$ *is fully grounded;*
- $A' = \sigma(A)$, $E' = \sigma(E)$, $C' = \sigma(C)$, $S' = \sigma(S)$ *and* $R' = \sigma(R)$*;*
- $T' = T$*;*
- $AgentID \in \mathcal{A}$ *is an agent identifier that corresponds to the agent affected by the norm, which is playing the target role* T*.*

For simplicity, we assume that once a deontic norm is being instantiated it is fully grounded. In order to ensure that all deontic instances have not free variables, all variables that occur in E, C, S, R may be contained in A (i.e., $v_A \supseteq v_E \cup v_C \cup v_S \cup v_R$ [3]).

For example, let us suppose that there is a heavy rain in an specific area a_1. Moreover, the *assistant* agent has been configured to obtain car routes. Thus, its user is a car driver and the Heavy Rain Norm is instantiated as follows:

$$\langle \mathcal{O}, slow(a_1), carDriver, user, heavyRain(a_1), \neg heavyRain(a_1), penalty, -\rangle$$

(Heavy Rain Instance)

3.3 Constitutive Norms

Legal codes do not only have normative prescriptions, but they also contain new definitions of categories and facts. This type of norms, which give an abstract meaning to facts, environmental elements, etc., is known as *constitutive norms*. Thus, they do not define restrictions on the behaviours. They introduce new classifications of facts and entities, called institutional facts [Sea69]. These abstract notions or facts have been traditionally used for the definition of general regulative norms [VS03, Ald09, AÁNDVS10]. For example *cheating* is an abstract fact that

[3] v_X is the set of variables occurring in any formula X

can be defined as *looking at a book in an exam* or *looking at others' cards in a card game*. The notion of *cheating* can be used in order to express in a single regulative norm that *all forms of cheating are forbidden*. According to this usage constitutive norms define the ontology used by the institution in the expression of regulative norms. Besides that, constitutive norms allow agents to know their capabilities for modifying the institutional state. Next, the formal definition of constitutive norms is provided.

3.3.1 Constitutive Norm Definition

Definition 3.3.1 (Constitutive Norm) *A constitutive norm n_c is a tuple $n_c = \langle I, A, E, BF, IF \rangle$ where:*

- $I \in \mathcal{I}$ *is the institution in which the constitutive norm is valid;*
- A, E *are wff of $\mathcal{L}$ that determine the norm validity period, i.e., they define the activation and expiration conditions, respectively;*
- $BF \in Lit(\mathbb{B})$ *represents the brute concept (brute fact) affected by the constitutive norm.*
- $IF \in Lit(\mathbb{I})$ *represents the institutional concept (institutional fact) defined by the constitutive norm.*

Those who are acquainted with works in the formalization of constitutive norms might be a little confused by the definition of both activation and expiration conditions for constitutive norms. The use of activation and expiration conditions allows us to provide a general description of the norm reasoning process. As a consequence, there is an agreement between the definitions of deontic and constitutive norms. However, it is possible to make a translation of our notion of constitutive norm into the well known definition provided by Searle [Sea69]. According to Searle, the form that constitutive norms take can be stated as the formula "BF *counts-as* IF in C" where the IF term is said to assign a new institutional definition or meaning to some brute fact BF. C represents the context or type of context in which the constitutive norm is applied. In our proposal this context is defined by means of the activation and expiration conditions (A and E, respectively). Thus, the activation and expiration conditions must be defined as C and $\neg C$ in order to model the fact that the constitutive norm is active only when C holds.

For example, in most countries like Spain, to drive exceeding the speed limits inside the town boundaries count-as a driving offence. In Spain this limit is defined as $50Km/h$. This fact is represented by the following constitutive norm:

$$\langle spain, inTown(T), \neg inTown(T), exceeed(50), drivingOffence\rangle \quad \text{(Driving Offence Norm)}$$

3.3.2 Constitutive Instance Definition

Once the activation conditions of a constitutive norm hold it becomes active and several constitutive instances, according to the possible groundings of the activation condition, must be created as follows:

Definition 3.3.2 (Constitutive Instance) *Given a theory Γ of $\mathcal{L}$ and a constitutive norm $c_n = \langle I, A, E, BF, IF\rangle$, an instance of c_n is defined as $i_c = \langle I', AgentID, A', E', BF', IF'\rangle$ where:*

- $\Gamma \vdash \sigma(A)$ *where σ is a substitution such as $\sigma(A)$ and $\sigma(E)$ are fully grounded.*
- $A' = \sigma(A), E' = \sigma(E)$, $BF' = \sigma(BF)$ *and* $IF' = \sigma(IF)$.
- $I' = I$
- $AgentID \in \mathcal{A}$ *is an agent identifier that corresponds to the agent affected by the norm, which belongs to the institution I.*

According to the previous definition instances may be partially grounded. Only A' and E' must be fully grounded, whereas BF' and IF' might have free variables. Therefore, $\mathcal{V}_A \supseteq \mathcal{V}_E$ to ensure that the expression E' has not free variables. Moreover, $\mathcal{V}_{BF} \setminus \mathcal{V}_A = \mathcal{V}_{IF} \setminus \mathcal{V}_A$ to allow institutional facts to be inferred from brute facts and vice versa.

For example, let us suppose that the *assistant* agent is located in Barcelona. Then the Driving Offence Norm is instantiated as follows:

$$\langle spain, user, inTown(barcelona), \neg inTown(barcelona), exceed(50), drivingOffence\rangle \quad \text{(Driving Offence Instance)}$$

3.4 Conclusions

In this chapter, the normative definitions used in this thesis have been provided. Specifically, the definitions of deontic and constitutive norms as well as deontic and constitutive instances have been provided. These definitions have been adapted from a previous model of norms for VO that has been proposed in [CAB09b, CAB11c]. The next chapters illustrate the agent architecture proposed in this thesis and how it allows the development of agents capable of reasoning about deontic and constitutive norms.

Chapter 4

The n-BDI Architecture

The first works on norms inside the MAS field assumed that agents are located in closed and relatively static systems where agents cooperate to achieve a common objective. For this reason, these first works were focused on programming norms inside the agent code. Later, the interest switched from such closed systems to open and dynamic systems in which heterogeneous and autonomous agents work together. Norm-programmed agents are unsuitable for these systems because of two main reasons [DMSC00]: the circumstances might change, which makes the programmed norms obsolete; and agents may interact with agents that follow different norms, in this situation explicit representations of norms can support appropriate, more flexible, reasoning. Thus, a shift from norm-programmed agents into norm-autonomous agents is necessary.

In [CCD99] a *norm-autonomous agent* is defined as an agent whose behaviour is influenced by norms that are *explicitly represented* inside its mind. Agents with an explicit representation of norms are able to belong to different societies, to communicate norms and to reason about them [LyLLd06]. Therefore, *norm-autonomous agents* have capabilities for *acquiring* norms; i.e., agents are capable of recognising the norms that are in force in their environment [AVC10]. Moreover, agents may have motivations to *accept* these recognised norms [LyLLd06]. For example, norms can be accepted since they have been promulgated by an authority. Besides that, agents are endowed with capabilities for determining whether a norm concerns their case and it is *relevant* [Kol05]. After the recognised norm has been accepted and considered as relevant, then agents must take the norm into account in their decisions. As mentioned in Section 2.6.2, despite the efforts that have been made to develop agents endowed with all of these capabilities, some important issues still pending. Specifically, Section 2.6.2 points out the

main deficiencies and drawbacks of the existing proposals on norm-autonomous agents.

With the aim of contributing to the resolution of these open problems, this thesis proposes a new architecture for norm-autonomous agents. This architecture, known as n-BDI, is an extension of a multi-context graded BDI architecture [CGS11] with an explicit notion of norm and instance. The n-BDI architecture allows modelling norm-autonomous agents that are endowed all the capabilities that norm-autonomous agents require. This chapter is organized as follows: Section 4.1 describes the running example used in this chapter; Section 4.2 details how the n-BDI architecture is defined as an extension of the multi-context graded BDI architecture with an explicit notion of norm and instance; Section 4.3 describes how norms are recognised and accepted; Section 4.4 describes how instances are created out of norms; Section 4.5 describes the experiment that we carried out; Section 4.6 contains the main contributions of this chapter; and Section 4.7 concludes this chapter.

4.1 Motivation Example

The first critical point of an architecture for norm-autonomous agents is why the explicit representation of norms is required. There are a lot of works that have proposed the use of norms for controlling MAS [BvdTV08a]. As argued in [CC95], there are two main ways in which norms can be implemented on agents: as *built-in* functioning rules and constraints or as *explicit mental objects* distinct from, say, goals and beliefs. The second alternative does not imply to model a norm-obeying system. Thus, normative reasoning must be performed on agents' mind even if the agents final decision is to transgress norms. There may be reasons supporting each one of these two alternatives. However, for dynamic and realistic environments, which are the ones considered by our proposal, where social actions are required the explicit representation of norms in agent minds is also necessary. For example, if deontic norms are implemented as hard-constraints on agents, then agents will follow blindly deontic norms. Thus, agents may be incapable of achieving their goals if the deontic norms are not well designed or the environment changes. Even if this extreme situation does not occur, the explicit representation of constitutive and deontic norms bring agents the possibility of belonging dynamically to unforeseen institutions. Moreover, the use of MAS for simulating realistic scenarios entails the development of social agents, in which normative reasoning is crucial. For these reasons the classic BDI architecture has been extended in this thesis with an explicit norm notion.

Along this thesis we will use an example to illustrate and motivate the need of the different elements that compose the norm autonomous architecture that is proposed in this thesis. Specifically, this example consists in a software agent, which will be named *assistant*, that draws up traffic routes according to the preferences that a human user has specified. These preferences may include time constraints, consumption requirements, avoidance of toll roads, and so on. Therefore the routes suggested by the *assistant* agent indicate not only the particular ways or directions but also the speed at which the human should drive at each stretch for meeting his requirements. Therefore, the *assistant* agent must consider as internal motivations (i.e., its desires) the user expressed preferences.

In order to calculate the most suitable route according to the user's preferences the *assistant* agent needs to know which are the norms that regulate the traffic in each region. If the suggested routes do not take into account norms, then the user that follows this route may be arrested and accused of a serious offence. Therefore, this scenario makes mandatory that software agents consider norms. These norms include both those formal norms that are defined explicitly in highway codes and those informal (i.e., social) norms that explain the attitude of the national population towards formal laws. There are some studies, such as the one made in [Bic06], sustaining the hypothesis that social norms or national culture are more important than formal laws in the attitude and behaviour of the driver population. The *assistant* agent may be used in different locations, in which the traffic is controlled by different norms; different users, which are influenced by different norms; or different times, in which the circumstances may change and the norms become obsolete; the explicit representation of norms supports appropriate and more flexible reasoning [DMSC00].

The *assistant* agent cannot consider norms as hard-constraints. For example, in some situations it might not be possible to find a traffic route that meets all the requirements and respects all deontic norms. In these situations, what is desirable is to find an equilibrium point between the user requirements, which are the internal motivations, and the deontic norms, which are the external motivations. Therefore, this case study will allow us to illustrate how our proposal allows software agents to consider and reason about both formal and social norms in a complex situation such as driving. The *assistant* agent does not take actions, it is only responsible for proposing the most adequate route to a human user who will determine if he/she follows the agent advice and performs the actions that have been suggested. Therefore, the *assistant* agent only is concerned about the reasoning processes that are performed before the

human user takes action. It allows us to focus on agent reasoning and normative reasoning in particular; forgetting other issues related to the performance of actions. Next, the agent architecture proposed in this thesis is described.

4.2 Normative Multi-context Graded BDI Architecture

Usually, proposals on agent architectures which support normative reasoning do not consider norms as dynamic objects which may be acquired and autonomously obeyed by agents [BDH$^+$01, ACCC08]. On the contrary, these proposals consider norms as static constraints that are programmed on agents. Therefore, agents are not able to deliberate about norms. The assumption that norms remain static makes sense from the institutional perspective. In this way, interactions among agents, which take place inside a specific institution, are controlled by a set of norms which remains quite stable. Due to this, institutional mechanisms for monitoring and enforcing norm compliance usually do not need capabilities for considering norms as dynamic entities. However, agents may join and leave several institutions along their execution. Therefore, the set of norms affecting them may change along time as they become members of different institutions. In addition, agents might belong simultaneously to different institutions which are controlled by conflicting norms. Thus, to have capabilities for explicit norm management and reasoning is a mandatory requirement for norm autonomous agents [CDJT00, CCD99].

As we mentioned before, in this thesis we propose a framework for allowing agents to consider norms in their decisions. The feature that distinguishes normative BDI agents from classic BDI agents is the availability of an explicit representation of norms and instances and the capabilities for reasoning about them. It serves this purpose well to address different mental attitudes in a modular way, and for that reason we rely on multi-context systems for the formalisation of those attitudes [GS94]. The main intuition beyond this kind of systems is that reasoning is usually performed on a subset of the global knowledge base. Each one of these subsets is a context. Informally, a context contains a partial theory of the world which encodes the agent's perspective about this part of the world. Each context has inference routines used to reason about it [Giu93]. Moreover, the reasoning in one context may affect reasoning in other contexts. Therefore, a multi-context system includes inference relationships among contexts.

Because we want our agents to contend with uncertainty and with conflicting mental states,

we will assume graded logics. As a consequence, in this thesis the multi-context graded BDI agent architecture [CGS11] has been extended with recognition and normative reasoning capabilities. According to the multi-context graded BDI agent architecture (proposed in [CGS11]), an agent is defined by a set of interconnected contexts or units $\langle \{C_i\}_{i \in I}, \Delta \rangle$. Each unit $c_i \in \{C_i\}_{i \in I}$ is a tuple $\langle Li, A_i, \Delta_i \rangle$, where L_i, A_i and Δ_i are the language, axioms and inference rules defining the logic of each unit, respectively. Δ is the set of bridge rules between the units; i.e., inference rules whose premises and conclusions belong to different contexts:

$$\frac{C_1 : A_1, ..., C_q : A_q}{C_j : A}$$

meaning that if for all $k \in \{1, ..., q\}$ A_k holds in C_k, then A is inferred in C_j. When a theory $\Gamma_i \subset L_i$ is associated with each unit, the specification of a particular agent is complete.

The multi-context graded BDI agent architecture does not provide an explicit representation of norms. However, it is capable of representing and reasoning with graded mental attitudes, that makes it suitable as a basis for the norm autonomous agent architecture. Consequently, the ***Normative Multi-context Graded BDI architecture*** (**n-BDI** for short) [CAB10a, CAB10b, CABN11] consists in extending the BDI architecture by adding new units and bridge rules in order to allow agents to make decisions with norms.

In this section we provide a general description of the **n-BDI** proposal, which is formed by: *mental* units to characterize beliefs (BC), intentions (IC) and desires (DC); *functional* units for planning (PC), communication (CC) and inferring reputation information (RC); and *normative* contexts for allowing agents to recognise new norms (NAC) and to consider norms in their decision making processes (NCC).

Following, each one of the contexts belonging to the n-BDI architecture is explained with more detail. In order to make a clear distinction among the different works that are the basis of the n-BDI proposal, each context will be described together with a reference where it was originally defined.

4.2.1 Mental Contexts

Mental contexts characterize beliefs (BC), intentions (IC), and desires (DC). All of them were initially defined in [CGS11] as units containing weighted propositions that represent the degree

of certainty, desirability, or intentionality of mental predicates.

- *Belief Context (BC).* It is formed by propositions belonging to the BC-Logic [CGS11]. The language $\mathcal{L}_{BC}$ is defined over a classical propositional language $\mathcal{L}$ (built from a countable set of propositional variables with connectives $\rightarrow$ and $\neg$) which is expanded with a fuzzy modal operator $\mathcal{B}$. Thus, it contains logic propositions such as $(\mathcal{B}\ \gamma, \rho)$, where $\mathcal{B}\ \gamma$ represents a belief about proposition $\gamma \in \mathcal{L}$ of an agent and $\rho \in [0, 1]$ represents the certainty degree associated to this belief. The logical connective $\rightarrow$ is used to represent explanation and contradiction relationships between propositions. Thus, $(\mathcal{B}\ \alpha \rightarrow \beta, \rho)$ represents that the agent believes that α explains β, with a certainty degree ρ. Similarly, expressions such as $(\mathcal{B}\ \alpha \rightarrow \neg\beta, \rho)$ means that the agent believes that proposition α contradicts proposition β, with a certainty degree ρ.

- *Desire Context (DC).* The original proposal of multi-context graded BDI [CGS11] defines a many value modal logic to represent and reason about agent bipolar preferences (i.e., positive and negative desires). For the purpose of this thesis, a single fuzzy modal operator $\mathcal{D}$ is required for representing desires. Thus, the DC contains logic propositions such as $(\mathcal{D}\ \gamma, \rho)$, where $\mathcal{D}\ \gamma$ represents a desire about proposition $\gamma \in \mathcal{L}$ of an agent and $\rho \in [0, 1]$ represents the desirability degree. Thus, negative desires are represented using the negation connective $\neg$ (i.e., $(\mathcal{D}\ \neg\gamma, \rho)$). Degrees of desires allow setting different levels of preference or rejection. The logical connective $\rightarrow$ is used to represent facilitation and incompatibility relationships between propositions. Thus, $(\mathcal{D}\ \alpha \rightarrow \beta, \rho)$ represents that proposition β, which can be either an action or a goal, achieves or facilitates proposition α in a degree ρ. Similarly, $(\mathcal{D}\ \alpha \rightarrow \neg\beta, \rho)$ implies that factor β is incompatible with factor α in a degree ρ.

- *Intention Context (IC).* It is formed by propositions belonging to the IC-Logic [CGS11]. Thus, it is formed by two kinds of graded intentions. The intention of a formula γ considering the execution of a particularly plan α is expressed as $(\mathcal{I}_\alpha\gamma, \rho_\alpha)$, where $\rho_\alpha \in [0, 1]$ may be considered as the truth degree of the expression *"γ is intended through plan α"*. The final intention to γ which takes into account the best plan to reach γ is denoted as $(\mathcal{I}\ \gamma, \rho)$. Thus, the intentionality degree of a proposition γ must be the consequence of finding a best feasible plan that permits a state of the world where γ holds to be achieved.

The logic of mental units is a mixture of first-order modal logic [Sha09], which is employed to represent those propositions that are believed, desired, or intended; and Rational Pavelka Logic (RPL) [Pav79] to represent the probability of propositions. Therefore, the axioms and rules are built by considering axioms of first-order predicate logic and axioms of RPL[1]. Deduction rules for each unit are Modus Ponens and Necessitation for the mental modalities $\mathcal{B}, \mathcal{D}, \mathcal{I}$. This thesis is not aimed at providing an exhaustive description of how the agent reasons about mental propositions. Therefore, only those aspects that are relevant to the norm reasoning process have been provided. For a complete description of these contexts see [CGS11].

4.2.2 Functional Contexts

The multi-context definition of a BDI agent [CGS11] proposes the definition of two functional contexts:

- The *Planner Context* (PC), which allows agents to decide the set of actions that will be attempted according to their desires. For the purpose of this thesis, the PC will be considered as a black box that builds plans to achieve the agent's desires, where plans have an associated cost according to the actions involved. Thus, the PC contains formulae such as $fplan(\gamma, \alpha, preC, postC, c_\alpha)$ that describe feasible plans for achieving γ . In particular, α is a set of actions that makes true γ; $preC$ and $postC$ are the plan preconditions and postconditions, respectively; and $c_\alpha \in [0, 1]$ is a real value that represents the cost of the plan.

- The *Communication Context* (CC) communicates agents with their environment. The theory inside the CC will take care of the sending and receiving messages to and from other agents. Thus it contains expressions such as $received(p, j, c)$ that represents those messages received by the agent. Specifically, p represents the illocution of a message [Sea69]; j is the identifier of the agent that has sent the message; c is the message content.

[1]RPL is an extension of Lukasiewicz's infinitely-valued logic by expanding its language with rational truth-constants to explicitly reason about degrees of truth [Háj98]. According to Lukasiewicz's logic, the following axioms are used throughout this thesis:

$$(\mathcal{M}\ \alpha, a), (\mathcal{M}\ \beta, b) \vdash (\mathcal{M}\ \alpha \wedge \beta, min(a, b))$$

$$(\mathcal{M}\ \alpha, a), (\mathcal{M}\ \alpha \rightarrow \beta, b) \vdash (\mathcal{M}\ \beta, a * b)$$

where $\mathcal{M} \in \{\mathcal{B}, \mathcal{D}, \mathcal{I}\}$ and $\alpha, \beta \in \mathcal{L}$

Besides that, in [PSMDP12], a reputation model was integrated into the multi-context graded BDI architecture. Mainly, this integration consists on adding the *Repage Context* (RC) for providing reputation information to the classic BDI reasoning process. The description of the Repage context is beyond the scope of this thesis. For the purpose of this thesis it is only necessary to know that agents are endowed with capabilities for evaluating performance of others (this subjective evaluation of a target is known as *image*) and exchanging this information (which is known as *reputation*). Thus, the RC contains reputation propositions such as $rep(j, r, \rho)$ that represents the reputation $\rho \in [0, 1]$ of agent j playing role r.

4.2.3 Normative Contexts

In order to extend classical BDI agents with an explicit notion of norm, the work of Sripada et al. [SS06] has been considered as a reference. It analyses the psychological architecture subserving norms. In particular, this architecture is formed by two closely linked innate mechanisms: one responsible for norm acquisition, which is responsible for identifying norm implicating behaviour and inferring the content of that norm; and the other maintains a database of those norms that are relevant to the current situation.

The norm reasoning problem assumes that norms are not initially implemented on agents' minds as constraints, but agents are able to acquire new norms and deliberate about them autonomously. In order to allow agents to have an explicit representation of norms and to consider them in their reasoning process, additional contexts in the BDI architecture are needed. Accordingly, the n-BDI proposal [dVCC$^+$10, CABN11] defines normative contexts for allowing an explicit representation and reasoning about norms:

- *Norm Acquisition Context (NAC).* It maintains a norm base that contains all norms which are applicable (i.e., *in force*) at a given moment. These norms are the external influences on agent's behaviour. Thus, external influences are represented inside the NAC, whereas internal motivations (such as goals) are represented in the DC. Section 4.3 provides a complete explanation of the NAC.

- *Norm Compliance Context (NCC).* This is the component responsible for reasoning about the set of norms which hold (i.e., that are relevant) at a specific moment. Section 4.4 provides a complete explanation of this context.

4.2.4 Reasoning Process in a n-BDI Agent

The reasoning process in a n-BDI agent is mainly performed by bridge rules that connect mental, functional and normative contexts. Thus, the information flows from perception to action via bridge rules that define how the information that is represented inside several contexts is combined for inferring new information in other contexts. The reasoning process can be summarised into three different phases. In the first one, the agent perceptions are used for updating the agent knowledge. In the second phase, desires are generated from the user preferences and norms. In the third phase, the agent makes a decision about the next action to be performed.

Phase 1. Perception. The agent perceives the environment and translates this perception into new formulae that are inserted in those contexts that are responsible for representing the agent environment. Specifically, new formulae are inserted into the BC, RC, NAC and NCC. The perception process is illustrated in Figure 4.1. This image shows how the different contexts (i.e., circles) are connected by means of bridge rules (i.e., boxes).

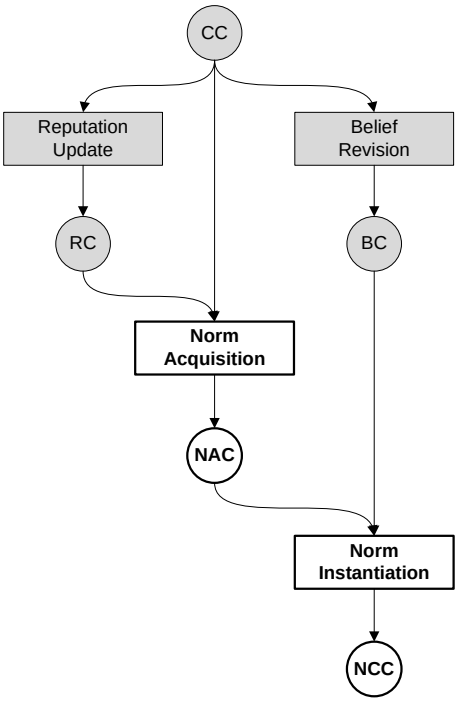

Figure 4.1: Perception Phase in the n-BDI Architecture. Contexts are represented as circles, whereas sets of bridge rules that perform similar tasks are represented as boxes in which there are input links, which are the premises of bridge rules, and output links, which represent the conclusions. Gray circles and boxes correspond to the basic architecture that has been defined in previous works [CGS11, PSMDP12]. The normative extensions are the white elements.

- *Belief Revision.* Belief revision is the process of changing beliefs to take into account a new piece of information [Han09]. For the moment being, neither the multi-context Graded BDI [CGS11] nor the n-BDI [dVCC$^+$10, CABN11] proposal have considered yet the problem of belief revision. However, a simple bridge rule for generating graded beliefs has been defined in [CGS11].

- *Reputation and Image Update.* As in case of the belief revision processes, the process by which the agent perceptions are used for updating the reputation and image of other agents is beyond the scope of this thesis. For the purpose of this thesis, it is only relevant to point out that the RC receives also all the messages received by the agent and infers reputation and image information from it. Specifically, the RC evaluates the behaviour of other agents forming their own image about the interacting participants. Moreover, the RC also considers the evaluations that are sent by agents (i.e., reputation messages). For a detailed description of this process see [PSMDP12].

- *Norm Acquisition and Acceptance.* It starts when the NAC receives information cues for inferring the norms that are applicable (i.e., *in force*) in the agent environment. Specifically, the NAC receives messages containing norm advices (i.e., information about the norms that regulate the agent environment) provided by other agents. Moreover, a n-BDI agent should make a decision about accepting or not these norm advices. In this work, the reputation of the informer agents is considered as a criterion for accepting or rejecting the norm advices. Therefore, several bridge rules have been defined to perform these tasks. Specifically, these bridge rules that infer formulae inside the NAC will be described in Section 4.3.2.

- *Norm Instantiation.* An agent considers a deontic norm as relevant when it believes that it is under the influence of this norm and it also believes that this norm is active. Constitutive norms become relevant when their activation condition holds. In these situations, deontic and constitutive norms must be instantiated and inserted into the NCC. Section 4.4 is focused in the NCC, so bridge rules that manage deontic and constitutive instances are explained in Section 4.4.2.

Phase 2. Deliberation. In this phase desires that represent the internal and external motivations of agents are created. The agent receives the user preferences as formulae

in the DC. Using a bridge rule the user's desires are transformed into graded positive and negative desires. Therefore, the user preferences are the internal source of agent's motivation, whereas the deontic norms are the external motivations.

- *Norm-based Expansion.* The norm-based expansion process propagates instances, currently in NCC, to the agent's mental and functional contexts through bridge rules. The consequences of instances are propagated to the agent's mental and functional contexts (i.e., the consequences of instances are *internalized*) every time NCC is updated because agent's actions are triggered by his prevalent state of mind. The "state of mind" is the union of the contents of all the contexts within a norm-aware agent. This includes normative elements. The norm-based expansion process depends on the type of norm that is being considered. Chapter 5 describes how this process takes into account deontic norms, whereas Chapter 6 describes how this process takes into account constitutive norms.
- *Coherence-based Contraction.* The internalization process just described may produce conflicts within each context. In those cases, the agent needs to address those conflicts so that he may take action. Specifically, our proposal employs *coherence* as a criterion for determining which propositions (both mental and normative) must be removed to resolve those conflicts. Specifically, we will profit from Joseph's proposal (it will be described in Section 7.3) to enable n-BDI agents to choose the propositions that maximize the coherence. In fact, we use coherence to face three different problems: (i) deliberating about the coherence of desires in view of applicable norms; (ii) determining degrees of coherence in states with normative conflicts; and (iii) in each context, choose a subset of maximal coherence to resolve normative conflicts. This process is explained in Chapter 7.

Phase 3. Decision Making. The decision making process is beyond the scope of this thesis, which is focused on the normative reasoning. However, we will like to illustrate how desires, which can be inferred from internal and external motivations, help the agent to select the most suitable plan to be intended and, as a consequence, normative actions might be carried out by the agent. An overview of the decision making phase is illustrated in Figure 4.2. For a detailed description of this process see [CGS11].

- *Plan Generation.* The most coherent desires and beliefs are passed from DC and BC

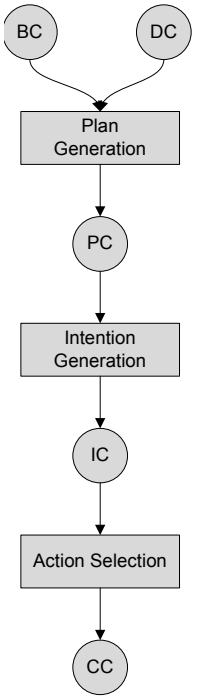

Figure 4.2: Decision-Making Phase in the n-BDI Architecture. Contexts are represented as circles, whereas sets of bridge rules that perform similar tasks are represented as boxes in which there are input links, which are the premises of bridge rules, and output links, which represent the conclusions. Gray circles and boxes correspond to the basic architecture that has been defined in previous works [CGS11, PSMDP12].

to PC. Then, the PC looks for feasible plans (i.e., plans that fulfil positive desires) that satisfy some preconditions and avoid undesired postconditions (i.e., negative desires). Thus, PC generates predicate instances of feasible plans (using the $fplan$ predicate).

- *Intention Generation.* Intentions to reach each positive desire are inferred by considering the trade-off between the benefit of reaching each desire and the cost of the best feasible plan that achieves them. This is done by the following bridge rule:

$$\frac{DC : (\mathcal{D}\ \gamma, \rho_{DC}), PC : fplan(\gamma, \alpha, preC, postC, c_\alpha)}{IC : (\mathcal{I}_\alpha \gamma, h(u(\rho_{DC}) - c_\alpha))}$$

where $u : [0, 1] \rightarrow \mathbb{R}$ is a non decreasing mapping that transforms desire degrees into negative costs(benefits); i.e., $u(\rho_{DC})$ can be interpreted as how much the user

accepts to pay to achieve a goal desired with degree ρ_{DC}, and $h : \mathbb{R} \to [0, 1]$ is a non-decreasing transformation that maps global benefits back to normalized utility degrees. Indeed, the value $h(u(\rho_{DC}) - c_\alpha)$ can be read as the monotone transformation of the expected benefit of intending γ through plan α.

- *Action Selection.* The previous bridge rule generates intentions for achieving desires through different plans. Inside the IC these intentions are considered for building final intentions to those desires. Thus expressions such as $(\mathcal{I}\ \gamma, \rho)$ represent the single intention for the desire γ. ρ is the intention degree of the best feasible plan for γ. Finally the PC and IC inform the CC of the best plan for each desire. This is done by the following bridge rule:

$$\frac{PC : fplan(\gamma, \alpha, preC, postC, c_\alpha), IC : (\mathcal{I}_\alpha\gamma, \rho), IC : (\mathcal{I}\ \gamma, \rho)}{CC : do(\alpha, \rho)}$$

The agent interacts with the environment through the CC by declaring which plan α the agent will finally execute. To do so, the CC selects the action with the highest degree among $do(\alpha, \rho)$ received via the previous bridge rule.

Since the main contribution of this chapter is the description of how n-BDI agents have an explicit representation of norms and instances, the following sections are focused on the NAC and NCC. Figure 4.3 illustrates this extension with more detail.

4.3 Norm Acquisition Context (NAC)

According to Conte et al. [CCD99], the problem of acquiring and recognising norms entails the evaluation of candidate norms against several criteria. For example, a deontic norm must be rejected if the agent that issues the norm is a non-recognised authority or if addressee agents are not within the scope of an authority. Thus, norm autonomous agents require capabilities for acquiring norms. In our proposal, the *Norm Acquisition Context* (NAC) allows agents to maintain a norm base that contains those norms which are applicable at a specific moment; i.e., the legislation that is in effect (*in force*) in a given moment. Specifically, the NAC receives information from the environment, determines whether that information is relevant to norms

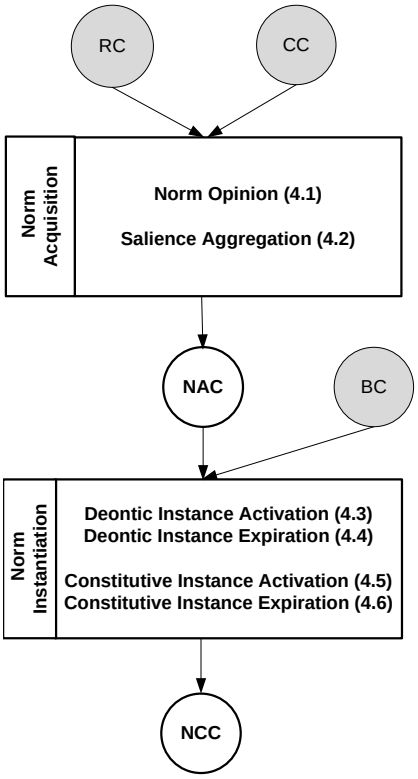

Figure 4.3: Representation of Norms and Instances in the n-BDI Architecture. In this image boxes have been split for showing the names of the bridge rules that perform each task.

that regulate the agent's environment and updates, accordingly, its existing set of norms. Thus, it is responsible for maintaining the set of norms that are applicable by acquiring the new norms and deleting the obsolete ones. This process can be defined as objective since no motivation or goal is considered in the acquisition process. Thus, agents only take into account their knowledge of the world for determining the set of norms which is more likely to be applicable.

For example, the *assistant* agent must be capable of planning routes across different regions and countries in which the traffic norms can be different. Moreover, traffic norms are occasionally modified. For example, the speed limits can be reduced in a specific road if many accidents take place in this road. For these reasons the *assistant* must be endowed with mechanisms that allow it to update the set of norms that regulate the traffic at each moment. Moreover, the fact that the *assistant* agent is capable of acquiring norms on-line implies a greater flexibility and a reduced load at the level of the agents' knowledge bases [CCD99].

Norm Recognition. Usually, computational models of autonomous norm recognition receive the agent perceptions, both observed and communicated facts, and identify the set of norms that control the agent environment. Perceptions which are relevant to the norm recognition

may be classified into:

- *Explicit normative perceptions.* They correspond to those messages exchanged by agents in which norms are explicitly communicated. Following this approach, several works have focused on analysing the role of leaders in the norm spreading. In particular, these leaders provide information about existing norms to follower agents [Ver00, SPP08].

- *Implicit normative perceptions.* This type of perceptions includes the observation of actions performed by agents as a way of detecting norms. Since deontic norms are usually supported by enforcing mechanisms such as sanctions and rewards, the detection of them has been considered as an alternative for acquiring new norms [FPU01]. Other works have proposed imitation mechanisms as a criterion for acquiring new deontic norms. These models are characterized by agents mimicking the behaviour of what the majority of the agents do in a given agent society [LyLLd06, CACC09]. Moreover, in [SA07] researchers have experimented with learning algorithms to identify a norm that maximizes an agent's utility.

- *Mixed normative perceptions.* There are proposals which consider both explicit and implicit normative perceptions as cues for inferring norms [ACCC08].

This work does not focus on the norm acquisition problem and the dynamics of norms. Here, we will only consider leadership-based norm spreading and the NAC will consider only *explicit normative information* (i.e., those messages exchanged by agents in which norms are explicitly communicated) as the only source of information for inferring norms. Besides that, the set of norms which are applicable may change both explicitly, by means of the addition, deletion or modification of the existing norms; and implicitly by introducing new norms which are not specifically meant to modify previous norms, but which change in fact the system because they are incompatible with such existing norms and prevail over them [GR08]. However, this is a complex issue which is out of the scope of this thesis. Works presented at the *Formal Models of Norm Change*[2] are good examples of proposals which provide a formal analysis of all kinds of dynamic aspects involved in systems of norms. For simplicity, we do not consider here the incompatibility relationships among norms.

[2]http://www.cs.uu.nl/events/normchange2/

Norm Salience. Agents are informed by expert agents (or *experts* for short) about the norms that are applicable at a specific moment. Specifically, agents are informed about the creation (*issuance*) and elimination (*abolition*) of norms that regulate their environment. Experts do not only inform about the issuance of norms, but also they are responsible for informing about the salience of norms. The norm salience is defined as the degree of activity and importance of a norm within a social group and a given context [CACC09]. As psychological [CRK90, Ban91] and behavioural economics [BC10, XH09] studies have pointed out, norm reasoning is strongly influenced by the salience of norms. Therefore, norm autonomous agents should be aware of salience of norms in order to make appropriate decisions about which norms to consider. For this reason, n-BDI agents represent norms together with their salience. However, the estimation of the norm salience is not trivial and it is beyond the scope of this thesis. The salience of norms can vary depending on social and individual factors. For example, the *surveillance rate* (frequency and intensity of punishment) is an important factor for determining the salience of deontic norms. Due to the difficulties that the determination of salience entails, we have assumed that the norm salience is estimated by experts which provide this subjective information to n-BDI agents.

For example, the *assistant* agent needs to represent and consider the salience of norms for deciding which deontic norms are less important and can be violated if necessary. As mentioned in the explanation of this case study, all traffic norms are not equally important. Moreover, the relative importance among traffic norms is a social factor that changes from one country to other. It may be argued that the importance of a traffic norm is implicitly represented in the strength of the sanctions (vs. rewards) that will be applied if any driver is caught transgressing this norm. However, this is not always true. For example, there are norms whose violations have similar sanctions but that are differently evaluated by the society. For example, usually the society considers that exceeding the speed limits as more reprehensible than driving without seatbelts. In some countries, like Spain, there are similar laws that forbid both behaviours. However, the traffic authorities invest more efforts on controlling and sanctioning to those drivers that do not respect the speed limits. As a consequence, the population considers that it is more important to obey speed limits norms since they are more frequently sanctioned. Moreover, there are specific moments (e.g., holidays) or facts (e.g., when the population is shaken by an accident that has made a great impact) that may affect the importance that the society and its control mechanisms (i.e., the traffic authorities) give to traffic norms.

4.3.1 NAC Language

4.3.1.1 Syntax

The NAC is a functional context that contains the set of applicable norms making use of two normative predicates: the *normOpinion* predicate, which is used for representing the salience that each expert assigns to a norm; and the *norm* predicate, which is used for representing the aggregated salience of a norm. The NAC is formed by expressions such as $norm(n, \rho)$, where n is a norm (for a formal definition of deontic and constitutive norms see Definitions 3.2.1 and 3.3.1, respectively) and $\rho \in [0, 1]$ is a real value that represents the salience of this norm. The NAC also contains expressions such as $normOpinion(n, j, \rho_j)$ where n is a norm, j identifies the expert that has provided the opinion and $\rho \in [0, 1]$ is the salience value that expert j has expressed for norm n.

4.3.1.2 Semantics

We define the semantics of the NAC language using operational semantics [Plo81]. Specifically, the operational semantics of the NAC is given by a set of rules that define a transition relation between configurations $\langle Opinion, Norm \rangle$ of the NAC where:

- *Opinion* is a set of norm opinions, where each opinion is an expression such as $normOpinion(n, j, \rho)$ that represents the salience (ρ) that an expert j assigns to a norm n.
- $Norm$ is a set of formulas where each formula is an expression such as $norm(n, \rho)$ that represents the aggregated salience (ρ) of a norm n.

The operational semantics for the NAC language formalises the transitions between possible configurations of the NAC. In the general case, in the agent's initial configuration both *Opinion* and $Norm$ are empty.

- *Norm Opinion Management.* The inference process of the NAC starts when a new norm opinion is generated (the process by which both norms and norm opinions are inferred in the NAC by means of bridge rules is described below in Section 4.3.2). Since this is the first opinion about a given norm provided by an expert, then the opinion is directly inserted into the NAC. Rule (a) in Table 4.1 represents this situation in which an expert provides its first opinion about a given norm. When other experts provide their first opinions about the same norm Rule (a) is executed again. When an expert provides

other opinions about the same norm, the norm opinion set is updated according to Rule (b) in Table 4.1.

- *Norm Management.* There are also operational rules that define the process by which the inferred norms are inserted inside the NAC. If a norm is inferred for the first time, then it is inserted into the NAC as indicated by Rule (c) in Table 4.1. If the same norm is deduced again, then the norm set is updated according to Rule (d) in Table 4.1.

$$\frac{normOpinion(N, J, \rho') \ \wedge \nexists normOpinion(N, J, \rho) \in Opinion}{\langle Opinion, Norm\rangle \longrightarrow \langle Opinion', Norm\rangle \quad Opinion' = Opinion \cup \{normOpinion(N, J, \rho')\}} \quad (a)$$

$$\frac{normOpinion(N, J, \rho') \wedge \exists normOpinion(N, J, \rho) \in Opinion}{\langle Opinion, Norm\rangle \longrightarrow \langle Opinion', Norm\rangle \quad Opinion' = Opinion \setminus \{normOpinion(N, J, \rho)\} \cup \ \{normOpinion(N, J, \rho')\}} \quad (b)$$

$$\frac{norm(N, \rho') \ \wedge \nexists norm(N, \rho) \in Norm}{\langle Opinion, Norm\rangle \longrightarrow \langle Opinion, Norm'\rangle \quad Norm' = Norm \cup \{norm(N, \rho')\}} \quad (c)$$

$$\frac{norm(N, \rho') \wedge \exists norm(N, \rho) \in Norm}{\langle Opinion, Norm\rangle \longrightarrow \langle Opinion, Norm'\rangle \quad Norm' = Norm \setminus \{norm(N, \rho)\} \cup \{norm(N, \rho')\}} \quad (d)$$

Table 4.1: Operational rules of the NAC Language

Both the syntax and the operational semantics of the NAC have been explained in this section. The bridge rules by which both opinions and norms are inferred in the NAC are explained below.

4.3.2 Norm Dynamics

As previously mentioned, expert opinions are considered for determining the salience of norms. Because the salience of a norm is a subjective information, it seems particularly appropriate to consider multiple experts, since multiple experts can provide more information than a single expert. An important issue that entails the use of multiple expert opinions is the aggregation of these opinions to produce a single combined opinion [CW99].

In this work, we have considered an appealing and simple approach to the aggregation of opinions: the *linear opinion pool* (LOP) [Sto61]. The linear opinion pool is just a weighted

linear combination of the experts' opinions. The weights in this approach can be used to represent the quality of the experts. The quality of experts is calculated by the RC, which determines the reputation of these agents as norm experts. The linear opinion pool satisfies some reasonable axioms such as the *unanimity* property (i.e., if all experts agree on an opinion the combined opinion also agrees). However, the linear opinion pool is prone to decision errors caused by "outlier" experts, since the arithmetic mean is not a robust estimator. As a solution to this problem, n-BDI agents use a *robust linear opinion pool* (R-LOP) [GP04] that reduces the effect of outlier experts in the aggregation of experts' opinions. This technique consists of a new formulation of the weights aimed at reducing the influence of "outlier" experts in the aggregation of opinions.

The process by which n-BDI agents update the norms and their salience is performed by a set of bridge rules that are applied any time the agent receives a message that informs about a change in the normative system (i.e., the set of norms that are applicable). Therefore, these bridge rules relate the communication context (CC) —through which messages are received— and the reputation information (contained in the RC) to the NAC, which contains the mental representation of norms. Next, norm acquisition bridge rules are defined: (i) norm opinion, and (ii) salience aggregation bridge rules.

4.3.2.1 Norm Opinion Bridge Rule

Communication related to the information about norms is considered for updating the NAC as follows (see Figure 4.3 Bridge Rule 4.1):

$$\frac{CC : received(inform, J, norm(\langle D, C, T, A, E, S, R\rangle, \rho))}{NAC : normOpinion(\langle D, C, T, A, E, S, R\rangle, J, \rho)} \tag{4.1}$$

If an agent is informed by another agent (the expert) J about the existence of a norm $(norm(\langle D, C, T, A, E, S, R\rangle), \rho)$, then this information must be employed for updating the NAC. ρ is the salience that the expert assigns to the norm. If the expert has not informed previously about this norm, a new opinion about this norm will be inserted inside the NAC $(normOpinion(\langle D, C, T, A, E, S, R\rangle, J, \rho))$ as indicated by Rule (a) in Table 4.1. Later, the expert might change the norm salience as indicated by sending other messages informing about the same norm. Thus, when an agent is informed by an expert agent J about the modification

of the salience of a norm i.e., $(norm(\langle D, C, T, A, E, S, R\rangle, \rho))$, then the opinion that is stored in the NAC is updated as indicated by Rule (b) in Table 4.1. Finally, the deletion of norms is represented as norms whose salience is 0 $(norm(\langle D, C, T, A, E, S, R\rangle), 0)$. When an agent is informed by an expert about the abolition of a norm, then the opinion provided by this agent must be updated accordingly. An expert considers that a norm has been abolished when it believes that the norm is not in effect and, as a consequence, its salience is equal to 0.

4.3.2.2 Salience Aggregation Bridge Rule

As previously stated, opinion from experts are considered for determining the salience of norms. Specifically, all opinions are aggregated following a robust linear opinion pool as follows (see Figure 4.3 Bridge Rule 4.2):

$$\frac{\begin{array}{c} NAC : normOpinion(\langle D, C, T, A, E, S, R\rangle, J_1, \rho_1), RC : (rep(J_1, normExpert, r_1)) \\ \ldots \\ NAC : normOpinion(\langle D, C, T, A, E, S, R\rangle, J_K, \rho_K), RC : (rep(J_K, normExpert, r_K)) \end{array}}{NAC : norm(\langle D, C, T, A, E, S, R\rangle, \theta_{acquisition})} \quad (4.2)$$

This bridge rule will be executed any time that an opinion changes or the reputation of an expert is modified. In this thesis, the reputation (r_i) of the informer agents as norm experts $(rep(J_i, normExpert, r_i))$ will be considered for determining the norm salience. Thus, the $\theta_{acquisition} \in [0, 1]$ aggregates opinions of experts by weighting these opinions. The robust aggregation of these opinions is described below.

Robust Linear Opinion Pool. This technique, which has been proposed in [GP04], first measures the conflict level introduced by every expert by taking into account the similarity between its opinion and reputation, and the other experts. Given a set of K probabilities $\Psi = \{\psi_1, ..., \psi_K\}$, where each $\psi_1 \in [0, 1]$; the similarity between one of the elements in Ψ and the other probabilities is defined as [TKS99]:

$$Sim_i(\Psi) = Sim(\psi_i, \Psi \setminus \{\psi_i\}) = 1 - \frac{1}{K-1} \sum_{j=1, j\neq i}^{K} |\psi_i - \psi_k|$$

Let us consider that there are K independent experts that express their opinion about salience of a given norm. Let $O = \{\rho_1, ..., \rho_K\}$, where each $\rho_j \in [0, 1]$, represents the set of opinions given by the different experts. Let $R = \{r_1, ..., r_K\}$, where each $r_j \in [0, 1]$, represents the set of reputations of the different experts. The conflict raised by an expert j ($j \in \{1, ..., K\}$) according to this opinion and its reputation is defined as:

$$Conflit_j = Sim_j(R)[1 - Sim_j(O)]$$

An expert who disagrees with the majority of other experts with a similar reputation is assumed to be conflicting (i.e., it is an "outlier" expert). Based on these conflict levels, the reliability of each expert j is calculated as follows:

$$Reliability_j = r_j(1 - Conflict_j)$$

where r_j is the reputation of expert j.

A reliable expert is the one that is both reputated and non-conflicting. Next, the aggregated opinion is obtained as the weighted average of the original expert opinions, with the weights being the reliability levels determined as before:

$$\theta_{acquisition} = \frac{\sum_{j=1}^{K} \rho_j * Reliability_j}{\sum_{i=1}^{K} Reliability_j}$$

As the experimental results in [GP04] illustrate, the robust linear opinion pool reduces or even cancels the negative influence of outlier experts.

For example, the *assistant* agent is informed about the Heavy Rain Norm (defined in Section 3.2.1) by three different experts. Each expert has its own opinion about the effectiveness of the Heavy Rain Norm. Therefore, the salience that each one gives to this norm is 0.25, 0.5 and 0.25. Therefore, $O = \{0.25, 0.5, 0.25\}$ is the set of opinions. The similarities between each one of the elements in O and the other two probabilities is $Sim(O) = \{0.88, 0.75, 0.88\}$. $R = \{0.75, 1, 1\}$ is the set of reputations of the three experts. The similarity among the reputations is $Sim(R) =$

$\{0.75, 0.88, 0.88\}$. The conflict raised by each expert is $Conflict = \{0.22, 0.09, 0.11\}$. Finally, the reliability of experts is $Reliability = \{0.59, 0.91, 0.89\}$. Therefore, the salience of this norm is 0.35[3] and the NAC contains a proposition such as:

$$norm(\langle \mathcal{O}, slow(A), carDriver, heavyRain(A), \neg heavyRain(A), penalty, - \rangle, 0.35)$$

4.4 Norm Compliance Context (NCC)

The *Norm Compliance Context* (NCC) is the component responsible for reasoning about the set of norms that hold at a specific moment. Thus, the NAC recognises all norms that are applicable, whereas the NCC only contains those norms which are active according to the current situation. The NCC is in charge of maintaining the set of instances that have been created out of the norms that are applicable (i.e., that are contained in the NAC).

For example, traffic norms are general norms that are not always active. Some of them, such as the Heavy Rain Norm, only come into effect under specific circumstances; e.g., when there is heavy rain. Therefore the *assistant* agent needs to be able to detect the activation and expiration conditions and update instances accordingly. In this case, the *assistant* might be informed by a server that provides meteorological information, so it can create or delete instances. What is considered as heavy rain is ambiguous. Therefore, this norm comes into effect under uncertain circumstances. Other examples of uncertainty conditions for norm activation are the constitutive norms that define what is considered as building work. When a road is being repaired there are several signals that inform the drivers about this fact. Frequently, these signals are not removed once the building work ends. In this situation the *assistant* agent must deal with a situation in which there are evidences, such as repairing signals, that sustain the building work situation; and evidences, such as the fact that no body is working on the road and the road seems in perfect state, that contradict this hypothesis. This section illustrates how n-BDI agents manage the activation and expiration of norms in uncertain environments.

[3]
$$\theta_{acquisition} = \frac{0.25 * 0.59 + 0.5 * 0.91 + 0.25 * 0.89}{0.59 + 0.91 + 0.89} = 0.35$$

4.4.1 NCC Language

4.4.1.1 Syntax

The NCC is a functional context that contains information about instances that have been built using the *instance* predicate. Thus, it contains expressions such as: $instance(i, \rho)$ where i is an instance (for a formal definition of deontic and constitutive instances see Definitions 3.2.2 and 3.3.2, respectively) and $\rho \in [0, 1]$ is a real value that represents the relevance degree of the instance. ρ can be interpreted as certainty about the activation of the norm.

4.4.1.2 Semantics

We define the operational semantics of the NCC language by a set of rules that define a transition relation between configurations $\langle Instance \rangle$ of the NCC where:

- $Instance$ is a set of instances, where each instance is an expression such as $instance(i, \rho)$ where i is an instance and ρ is the certainty degree of the instance.

The operational semantics for the NCC language formalises the transitions between possible configurations of the NCC. In the general case, an agent's initial configuration is $\langle Instance \rangle$ where $Instance$ is empty.

The reasoning cycle starts when a new instance is generated (the process by which instances are inferred in the NCC is described below in Section 4.4.2). Since this is the first time that an instance is deduced, it is inserted into the NCC. Rule (a) in Table 4.2 represents this situation in which instances are inferred in the NCC for the first time. If an instance that already belongs to the NCC is deduced again, then the instance set will be updated according to Rule (b) in Table 4.2.

$$\frac{instance(I, \rho) \ \wedge \nexists instance(I, \rho') \in Instance}{\begin{array}{c}\langle Instance \rangle \longrightarrow \langle Instance' \rangle \\ Instance' = Instance \cup \{instance(I, \rho)\}\end{array}} \quad \text{(a)}$$

$$\frac{instance(I, \rho) \wedge \exists instance(I, \rho') \in Instance}{\begin{array}{c}\langle Instance \rangle \longrightarrow \langle Instance' \rangle \\ Instance' = Instance \setminus \{instance(I, \rho')\} \cup \{instance(I, \rho)\}\end{array}} \quad \text{(b)}$$

Table 4.2: Operational rules of the NCC Language

The language that allows instances to be represented in the NCC has been explained in this section. The bridge rules by which deontic and constitutive are inferred inside the NCC are explained below.

4.4.2 Instance Dynamics

As stated before, norms are not always active. Thus, instances are created out of the norms when the activation condition holds. Agents must have beliefs that sustain the activation of norms in order to create instances. Similarly, norms also include an expiration condition that defines the validity period or deadline of instances. Thus, agents must believe that a given instance has expired in order to delete its mental representation. Therefore, instance dynamics consists on mental processes for creating and deleting instances. These two processes have been defined by means of bridge rules that relate the agent beliefs to the mental representation of norms and instances. These bridge rules depend on the type of the norm that is being considered. Next, bridge rules for detecting the activation and expiration of deontic and constitutive norms are explained.

4.4.2.1 Dynamics of Deontic Instances

Deontic Instance Activation Bridge Rule. Deontic norms are instantiated in the agent mind when the agent believes the activation condition to be true and it also believes that it is under the influence of the deontic norm; i.e., it enacts the target role of the norm (see Figure 4.3 Bridge Rule 4.3):

$$\frac{\begin{array}{c} NAC : norm(\langle D, C, T, A, E, S, R\rangle, \rho_{NAC}), \\ BC : (\mathcal{B}\, A', \rho_{A'}), BC : (\mathcal{B}\, play(AgentID, T'), \rho_{T'}) \end{array}}{NCC : instance(\langle D, C', T', AgentID, A', E', S', R'\rangle, f_{relevance}(\rho_{A'}, \rho_{T'}))} \tag{4.3}$$

If an agent considers that a deontic norm ($\langle D, C, T, A, E, S, R\rangle$) is currently active —i.e., it is believed that the activation condition holds $\Gamma_{BC} \vdash (\mathcal{B}\, A', \rho_{A'})$— and the agent considers that it is under the influence of the deontic norm —i.e., $\Gamma_{BC} \vdash (\mathcal{B}\, play(AgentID, T'), \rho_{T'})$[4] — then a new deontic instance is generated[5].

[4] *play* is a binary predicate that models the enactment of roles. Specifically, the expression $play(a, r)$ describes the fact that the agent identified by $a \in \mathcal{A}$ plays the role identified by $r \in \mathcal{R}$.

[5] Deontic instances are created independently of the agent that is executing this reasoning process. It allows

The relevance degree assigned to the deontic instance is defined by the $f_{relevance}$ function, which combines the amount of belief about the activation of the deontic norm (i.e., the certainty degree $\rho_{A'}$) and the certainty about the norm that affects the agent ($\rho_{T'}$) to update the certainty of an instance. Therefore, it is defined as a numerical fusion operator[6] that can be given different definitions depending on the properties that are required in each concrete application. In case of instances, there are two beliefs (i.e., $(\mathcal{B}\,A', \rho_{A'})$ and $(\mathcal{B}\,play(AgentID, T'), \rho_{T'})$) that are required to confirm the instantiation and relevance of norms. Specifically, if there is a high certainty about these two conditions, then the deontic instance must have a higher relevance. Similarly, if there is a low certainty about these two conditions, then the deontic instance must have a lower relevance. As a consequence, the combination among the uncertain values that cause the norm internalization is defined as a symmetric sum as follows:

$$f_{relevance}(\rho_{A'}, \rho_{T'}) = \frac{\rho_{A'} * \rho_{T'}}{1 - \rho_{A'} - \rho_{T'} + (2 * \rho_{A'} * \rho_{T'})}$$

Therefore, the $f_{relevance} : [0,1]\times[0,1] \rightarrow [0,1]$ is a function such that [DP85]: $f_{relevance}(0,0) = 0$ and $f_{relevance}(1,1) = 1$, $f_{relevance}$ has as null element 0, $f_{relevance}$ is increasing with respect to both arguments and continuous. Symmetrical sums represent variable aggregation operators; i.e., the behaviour of the operator depends on the values that are combined:

- $f_{relevance}(\rho_{A'}, \rho_{T'}) \leq min(\rho_{A'}, \rho_{T'})$ if $max(\rho_{A'}, \rho_{T'}) < 0.5$. This behaviour is known as conjunctive or severe, since it provides a combined result which is lower than each individual information.

- $f_{relevance}(\rho_{A'}, \rho_{T'}) \geq max(\rho_{A'}, \rho_{T'})$ if $min(\rho_{A'}, \rho_{T'}) > 0.5$. This behaviour is known as disjunctive or indulgent, since it provides a combined result which is higher than each individual information.

- $x < f_{relevance}(\rho_{A'}, \rho_{T'}) < y$ (or $y < f_{relevance}(\rho_{A'}, \rho_{T'}) < x$) if $x \leq 0.5 \leq y$ (or $y \leq 0.5 \leq x$). This behaviour is known as cautious, since it provides a combined result which is a compromise between the individual information.

For example, let us suppose that the *assistant* agent is informed by a meteorological server that there is a heavy rain in an specific area a_1 with a 75% of probability. Moreover, the

n-BDI agents to be aware of which deontic norms affect other agents, which can be useful for predicting and evaluating the behaviour of its interaction partners. However, this issue is beyond the scope of this thesis.

[6]For a review and classification of data fusion operators see [Blo96].

assistant agent has not been configured to obtain car routes and it assumes that the human user is a car driver with a 50% of probability. Therefore, the Deontic Instance Activation Bridge Rule is applied as follows:

$$\frac{\begin{array}{c}NAC : norm(\langle \mathcal{O}, slow(A), carDriver, heavyRain(A), \neg heavyRain(A), penalty, -\rangle, 0.35), \\ BC : (\mathcal{B}\, heavyRain(a_1), 0.75), BC : (\mathcal{B}\, play(self, carDriver), 0.5)\end{array}}{\begin{array}{c}NCC : instance(\langle \mathcal{O}, slow(a_1), carDriver, self, heavyRain(a_1), \neg heavyRain(a_1), \\ penalty, -\rangle, f_{relevance}(0.75, 0.5))\end{array}}$$

The relevance of the instance is 0.75[7] and the NCC contains a proposition such as:

$$\begin{array}{c}instance(\langle \mathcal{O}, slow(a_1), carDriver, self, heavyRain(a_1), \neg heavyRain(a_1), \\ penalty, -\rangle, 0.75)\end{array}$$

Deontic Instance Expiration Bridge Rule. Once the expiration condition of a deontic instance holds, then the certainty of the instance is reduced (see Figure 4.3 Bridge Rule 4.4):

$$\frac{\begin{array}{c}NCC : instance(\langle D, C', T', AgentID, A', E', S', R'\rangle, \rho_{NCC}), \\ BC : (\mathcal{B}\, E', \rho_{E'})\end{array}}{\begin{array}{c}NCC : instance(\langle D, C', T', AgentID, A', E', S', R'\rangle, \\ f_{expiration}(\rho_{NCC}, \rho_{E'}))\end{array}} \tag{4.4}$$

If the NCC of an agent contains a deontic instance ($instance(\langle D, C', T', AgentID, A', E', S', R'\rangle, \rho_{NCC})$) and it has a belief that sustains its expiration ($\mathcal{B}\, E', \rho_{E'}$), then the degree of the instance must be reduced. In case of the expiration of norms the belief ($\mathcal{B}\, E', \rho_{E'}$) disconfirms with the instance. Thus, we have considered the rules of MYCIN for combining certainty factors [SB75]. These "certainty factors" take their values within the $[-1, 1]$ interval; i.e., they are positive if confirm an event and negative if the information disconfirms the event. Therefore, MYCIN operators have been selected as a basis since they allow evidences that confirm or disconfirm an event or hypothesis to be combined. According to the rules defined

[7] $$f_{relevance}(0.75, 0.5) = \frac{0.75 * 0.5}{1 - 0.75 - 0.5 + (2 * 0.75 * 0.5)} = 0.75$$

in [SB75] for combining disconfirming information, the $f_{expiration}$ function is defined as follows:

$$f_{expiration}(\rho_{NCC}, \rho_{E'}) = max(0, \rho_{NCC} - \rho_{E'})$$

Therefore, the $f_{expiration} : [0,1] \times [0,1] \rightarrow [0,1]$ is a function such that [SB75] the unit element is 0, which is an information that says nothing and does not influence the combination. Thus, if there is a high certainty about the expiration of the instance, the relevance degree of the instance would become 0. In this case, the instance would be removed from the NCC and no longer considered by the decision making process.

4.4.2.2 Dynamics of Constitutive Instances

Constitutive Norm Activation Bridge Rule. Constitutive norms are instantiated in the agent mind when the agent believes the activation condition to be true (see Figure 4.3 Bridge Rule 4.5):

$$\frac{\begin{array}{c} NAC : norm(\langle I, A, E, BF, IF\rangle, \rho_{NAC}), \\ BC : (\mathcal{B}\, A', \rho_{A'}), BC : (\mathcal{B}\, member(AgentID, I'), \rho_{I'}) \end{array}}{\begin{array}{c} NCC : instance(\langle I', AgentID, A', E', BF', IF'\rangle, \\ f_{relevance}(\rho_{A'}, \rho_{I'})) \end{array}} \tag{4.5}$$

If an agent considers that a constitutive norm ($\langle I, A, E, BF, IF\rangle$) is currently active —i.e., it is believed that the activation condition holds $\Gamma_{BC} \vdash (\mathcal{B}\, A', \rho_{A'})$— and the agent considers that it is under the influence of the constitutive norm —i.e., $\Gamma_{BC} \vdash (\mathcal{B}\, member(AgentID, I'), \rho_{I'})$[8] — then a new instance is generated—i.e., the formula ($instance(\langle I', AgentID, A', E', BF', IF'\rangle, f_{relevance}(\rho_{A'}, \rho_{I'}))$) is inserted in the NCC—.

The degree assigned to the instance is defined by the $f_{relevance}$ function which combines the amount of belief about the activation of the constitutive norm (i.e., the certainty degree $\rho_{A'}$) and the certainty in which the norm affects the agent (i.e., the certainty degree $\rho_{I'}$).

Constitutive Instance Expiration Bridge Rule. Once the expiration condition of a constitutive instance holds, then the certainty of the instance is reduced (see Figure 4.3 Bridge

[8] *member* is a binary predicate that models the institution membership. Specifically, the expression $member(a, i)$ describes the fact that the agent identified by $a \in \mathcal{A}$ belongs to the institution $i \in \mathcal{I}$.

Rule 4.6):

$$\frac{NCC : instance(\langle I', AgentID, A', E', BF', IF' \rangle, \rho_{NCC}), BC : (\mathcal{B}\, E', \rho_{E'})}{NCC : instance(\langle I', AgentID, A', E', BF', IF' \rangle, f_{expiration}(\rho_{NCC}, \rho_{E'}))} \tag{4.6}$$

If the NCC contains a constitutive instance and there is a belief that sustains its expiration, then the degree of the constitutive instance must be reduced. Therefore, the belief $(\mathcal{B}\, E', \rho_{E'})$ disconfirms with the constitutive instance.

4.5 Acquiring Norms: Experimental Results

In this section, we describe the experiment that we carried out to experimentally evaluate the performance of n-BDI agents when they acquire norms; i.e., when they recognise norms and determine their importance. As explained in Section 4.3.2 n-BDI agents consider multiple opinions for determining the salience of norms. Specifically, these opinions are combined by using the robust linear opinion pool (R-LOP). However, other techniques can be used to aggregate the opinions of experts. For example, the linear opinion pool (LOP) [Ber85] aggregates the opinions as a weighted mean. A simpler approach consists in considering the opinion of a single expert. Specifically, only the opinions of the best expert (BE) (i.e., the most reputed expert) are taken into account. Thus, in this experiment we compare the performance of n-BDI agents when they use the R-LOP, LOP and BE techniques to calculate the salience of norms. Specifically, we have compared the relative error made by n-BDI agents on average when they calculate the salience of norms. Let the salience of a norm be ρ and the salience estimated by an agent i be ρ^i. Then the relative error is defined by:

$$\frac{|\rho^i - \rho|}{\rho}$$

Given a set of N norms (the real salience of any norm j is denoted by ρ_j) and a set of A agents (the salience that any agent i estimates about a norm j is denoted by ρ_j^i), the average relative error made by these agents when they calculate the salience of these norms is defined by:

$$\frac{\sum_{i=1}^{A} \sum_{j=1}^{N} \frac{|\rho_j^i - \rho_j|}{\rho_j}}{A * N}$$

Parameter	Value
# of norms	100
# of agents	100
# of simulations	1000
# of experts	$[5, 50]$
Expert accuracy	$[0, 1]$
Agent accuracy	$[0, 1]$

Table 4.3: Parameters used in the norm recognition experiment

We considered a scenario with the parameters that we sum up in Table 4.3. In this scenario, we employed 100 agents. These agents belong to the same institution in which there are 100 different norms. Agents are informed by a set of experts about these norms. Specifically, the number of experts ranges from 5 to 50 in the experiment. The accuracy of each one of the experts to determine the salience of norms ranges randomly within the $[0, 1]$ interval. The higher the accuracy of an expert, the more precise the opinions that the expert provides. Hence, the opinions provided by experts are affected by a random normally-distributed noise. We consider a normally-distributed noise with mean 0.0 and a varying standard deviation depending on the expert accuracy[9]. Finally, n-BDI agents should determine which is the reputation of each expert with respect to their recommendations about norms. Each n-BDI agent has an accuracy degree that ranges within the $[0, 1]$ interval and determines the exactness of the reputations that it calculates. Reputations are also affected by a random normally-distributed noise.

In each simulation, agents are created with a random accuracy degree. Moreover, a set of experts, which have a random accuracy, is also created. Agents ask all experts about the salience of norms. Each expert provides each agent with a different opinion for each norm[10]. According to the opinions provided by experts and the reputations that each agent assigns to experts, the salience of norms is calculated by agents. Each simulation has been repeated 1000 times to support findings. Table 4.4 shows the relative error that agents made on average with respect to the number of experts. Regardless of the number of experts R-LOP agents perform better. As the number of experts decreases, the difference between R-LOP and LOP becomes lightly smaller. In these scenarios is more difficult to select outlier experts and R-LOP agents behave as LOP agents.

[9]Specifically, we consider the distribution $\mathcal{N} \sim (0, \frac{1-accuracy}{2})$.

[10]Experts do not provide always the same opinion about a given norm; i.e., they estimate the salience of norms each time they are asked by agents. However, the error made by each expert when it estimates the salience of any norm is bounded by its accuracy degree.

# of experts	R-LOP	LOP	BE
5	$17.43 \pm 0.36\%$	$18.09 \pm 0.36\%$	$22.85 \pm 0.75\%$
10	$12.25 \pm 2.98\%$	$13.04 \pm 3.08\%$	$18.09 \pm 7.88\%$
15	$10.12 \pm 2.22\%$	$10.95 \pm 2.36\%$	$16.39 \pm 5.85\%$
20	$9.04 \pm 1.75\%$	$9.91 \pm 1.91\%$	$15.98 \pm 5.32\%$
25	$8.1 \pm 1.49\%$	$8.97 \pm 1.64\%$	$15.29 \pm 4.69\%$
30	$7.52 \pm 1.28\%$	$8.42 \pm 1.45\%$	$14.86 \pm 4.28\%$
35	$7.13 \pm 1.19\%$	$8.04 \pm 1.35\%$	$14.72 \pm 4.32\%$
40	$6.77 \pm 1.09\%$	$7.71 \pm 1.27\%$	$14.77 \pm 3.97\%$
45	$6.51 \pm 1.07\%$	$7.47 \pm 1.26\%$	$14.63 \pm 3.98\%$
50	$6.22 \pm 1\%$	$7.19 \pm 1.18\%$	$14.55 \pm 4.26\%$

Table 4.4: 95% confidence interval for the relative error made by agents

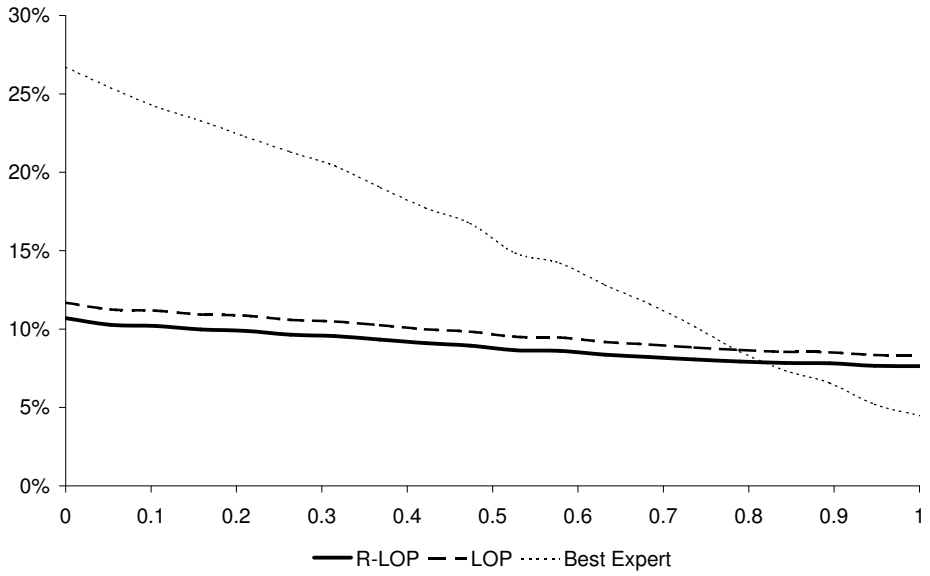

Figure 4.4: Relative error with respect to the agent accuracy

Figure 4.4 shows the relative error made by agents with respect to the agent accuracy. Specifically, this figure shows the results obtained when there are 10 experts. When the accuracy of agents is lower than 0.9, then BE agents cannot determine which is the best expert and they are unable to calculate the salience of norms properly. When the accuracy of agents is very high, then the performance of BE agents is as good as or even better than R-LOP. As illustrated by Table 4.4, BE agents obtain worse results on average. Thus, it is better to consider multiple opinions when the accuracy of agents takes random values. As illustrated by Table 4.4, R-LOP agents acquire norms more precisely than LOP agents. However, as the agent accuracy increases the difference among the results obtained by R-LOP and LOP agents decreases lightly since the reputation value is precise enough to determine which outlier experts are.

In light of the results of this experiment, we can conclude that: (i) n-BDI agents are able to acquire norms with reasonable quality (the relative error made by n-BDI agents is lower than 18% considering only 5 experts with random accuracies); (ii) agents that take multiple opinions into account (R-LOP and LOP agents) obtain better results than agents that do not consider multiple opinions (BE); and (iii) n-BDI agents are capable of identifying outlier experts minimising their influence.

4.6 Contributions

The main contributions of the n-BDI architecture proposed in this chapter are:

- *Norm Representation*: n-BDI agents are capable of representing both norms and their instances thanks to the NAC and NCC contexts. It may be arguable the need of these two contexts:
  - MAS research has given different meanings to the norm concept. For example, it has been employed as a synonym of obligation and authorization [Dig99], social law [MT95], social commitment [Sin99] and other kinds of rules imposed by societies or authorities. The n-BDI proposal is based on the notion of norm as an abstract rule that defines under which circumstances norms are instantiated. This notion of norm has been widely used by other relevant works in norm-autonomous agents such as [OLMN08, LyLLd06, Kol05]. Therefore, its formal definition, semantics and dynamics is well-known. The definitions of norm and instance have been particularized into the notions of deontic and constitutive norm and instance. As far as we are concerned, this is the first proposal of norm-autonomous agent that considers constitutive norms explicitely.
  - Since the main aim of this chapter is to illustrate how n-BDI agents take norms into account in their reasoning process we have extended the multi-context BDI architecture with two normative contexts (i.e., the NAC and the NCC) for representing norms and instances. We have decided to represent norms and instances separately in the NAC and NCC due to two main reasons. Firstly, we consider that representing norms and instances independently of other mental propositions allows to explain the norm reasoning with more clarity. Thus, the NAC and the NCC allows us to

define explicitly the relationships among norms, instances and the other contexts. Moreover, the explicit distinction between instances and norms allows to illustrate the differences between them; i.e., they have a different definition, semantics and dynamics and are considered in different steps of the reasoning process. The second reason for the creation of the NAC and NCC contexts is the fact that norms and instances are different from beliefs. Norms and instances are not simply beliefs since they entail processes for accepting them, determining their relevance and deciding about norm compliance. These processes do not occur in case of beliefs. Norms and instances are the external motivations of agents [CC95, DKS02] and we have considered that it is more suitable to represent them independently from beliefs and desires.

- *Norm Acquisition*: In the n-BDI architecture agents are capable of acquiring the set of norms that regulate their environment as well as determining the salience of these norms.
  - Specifically, n-BDI agents acquire norms by considering those messages in which explicit information about the applicable norms is provided by other agents (i.e., experts). This norm acquisition mechanism has been selected since it is quite simple and allows us to avoid the complex issue of norm learning, which is beyond the scope of this work. Moreover, this norm acquisition mechanism is compatible with several MAS frameworks and infrastructures in which norms are stored in public repositories or components (e.g., the OMS in the THOMAS framework [CJBA10]) or artifacts (e.g., the NormativeBoard in the ORA4MAS framework [HBKR10b]).
  - As argued in Section 4.3, an important factor when humans reason about norms is their salience (i.e., the importance of these norms). Salience of norms is also important in MAS; e.g., there may be a hierarchy of norms in which some norms are defined as more important than others. For this reason, n-BDI agents not only have capabilities for acquiring the set of norms that are applicable in their environment, but they also are capable of acquiring the salience of these norms.
- *Norm Acceptance*: n-BDI agents represent all norms that have been acquired inside the NAC. It may seem that all norms are automatically accepted by agents. However, the degree in which the norms influence the agent behaviour depends on the norm salience. Therefore, n-BDI agents accept norms when they consider them as salient in their society.

  - In the n-BDI proposal the norm salience is determined by considering the opinions of experts. This mechanism has been selected since it allows us to refrain from developing procedures for learning the norm salience. Besides that, the n-BDI proposal considers the opinion of multiple experts, since multiple experts can provide more information than a single expert.
  - The opinion of experts has been aggregated to produce a single combined salience value. Specifically, the salience opinions are combined considering the reputation of each expert using a robust aggregation operator that reduces the impact of outlier experts.

- *Norm Relevance*: n-BDI agents consider a given norm as relevant to their case when they are under its influence and the norm is active.

  - In the n-BDI proposal, as in case of other well-known proposals on norm-autonomous agents [LyLLd06, OLMN08, Kol05], activation and expiration conditions have been used to define the period in which norms come into effect. However, all of the previous proposals do not consider that agents have an uncertain knowledge of the world. Therefore, only the n-BDI proposal confronts with the activation and expiration of norms within uncertain environments.
  - Besides that, in the n-BDI proposal the notion of role has been used to define the sphere of influence of deontic norms. The use of deontic norms for defining the responsibilities, duties and rights of roles has been proposed also in other works such as [LyLLd06, OLMN08, DVSD05]. Similarly, institutions define the scope of constitutive norms.
  - n-BDI agents consider a given norm as relevant to their case when they are under its influence and the norm is active. Specifically, n-BDI agents combine the certainty values assigned to these two facts by a dynamic fusion operator that takes into account the values that are combined.

4.7 Conclusions

In this chapter we have explained the extension of a BDI architecture for allowing agents to have an explicit representation of norms and instances. Thus, agents are capable of representing the

norms that are applicable in their environment as well as detecting which ones are active at a given moment. This chapter focuses on the perception phase in which agents used update their beliefs and determine the norms and the instances that are relevant to their current situation. However, the problem of how agents take them into account has not been considered yet. The next chapters propose deliberative processes for considering deontic and constitutive norms in the n-BDI architecture.

Chapter 5

Norm-based Expansion: Reasoning About Deontic Norms

n-BDI agents require capabilities for acquiring and accepting norms, determining when norms are relevant to their case and deciding which ones will be obeyed. In this chapter, we propose a procedure for making decisions about norm compliance based on three different factors: self-interest, enforcement mechanisms and internalised emotions. Different agent personalities can be defined according to the importance given to each factor. These personalities have been experimentally compared and the results are shown in this chapter. This chapter is structured as follows: Section 5.1 contains an introduction to this chapter; Section 5.2 describes the process of norm-based expansion for deontic norms; Section 5.3 describes the functions that allow n-BDI agents to make decisions about norm compliance; Section 5.4 describes the experiment that has been carried out; Section 5.5 summarises the main contributions of this chapter; and Section 5.6 concludes this chapter.

5.1 Introduction

Despite the efforts that have been made to develop agents endowed with capabilities for taking into account norms in their decisions, some important issues are still pending. The best to our knowledge, the open issue that has received the least attention is the development of procedures for making autonomous decisions about norm compliance. Up to now, the decisions about norm compliance consider the effects of violating and obeying norms on the agent goals for making decisions about norm compliance [AVC10, BL01, LyLLd06]. However, there are works on

the psychology field [Els89, Els00] that claim that norm compliance is not only explained by rational reasons that consider the impact of norms and their enforcement procedures (sanctions and rewards) on the agent's goals. Besides that, there are emotional reasons, which are related to emotions, such as shame, that have not been considered yet in the development of norm-autonomous agents.

This chapter answers two main questions: "Is it possible to develop norm-autonomous agents that take into account the emotional repercussion of norms when they make decisions about norm compliance?", and "Does it make sense that software agents take into account these emotional factors?". In response to the first question, in this chapter we define a set of functions that allow agents to determine their willingness to comply with deontic norms according to rational and emotional factors. In response to the second question, we have developed an experiment for illustrating the performance of these functions. The results show that the emotional factors sustain compliance with more norms than rational factors.

5.2 Norm-based Expansion for Deontic Norms: Norm Internalization

As stated before, the norm-based expansion consists in extending the agent "state of mind" accordingly to instances. This process is known as *internalization*. Maybe the most relevant proposal on the norm *internalization* in MAS is the work of Conte et al. [CAC10]. According to them, a characteristic feature of norm internalization is that norms become part of the agent's identity; i.e., norms become part of the cognitive elements of the individual agent. In this thesis a simple approximation to the norm internalization process has been considered. In particular, we have only considered the internalization of norms as goals. In this sense, the process of norm internalization has been described by the self-determination theory [DR00] as a dynamic relation between norms and desires. This shift would represent the assumption that internalised norms become part of the agent's sense of identity. In future extensions of this architecture, we will consider the internalization of norms as beliefs and intentions.

In the running example used in this theis, the *assistant* agent needs a mechanism to decide to what extent deontic norms that regulate traffic will be respected in the proposed routes. This mechanism should consider the importance of norms, the certainty about its activation and the user preferences. This section illustrates how a general purpose n-BDI agent faces with

this complex issue.

As claimed in [AVC10] "*usually normative beliefs generate normative goals*". Thus, after performing the instantiation process for creating new instances, the NCC must update the DC with the new normative desires. These new desires derived from instances may trigger the creation of new intentions. Besides that, they may help the agent to select the most suitable plan to be intended and, as a consequence, normative actions might be carried out by the agent. As illustrated by Figure 5.1, the *Norm Internalization* bridge rules relate relevant instances (contained in the NCC) with the agent beliefs (contained in the BC), desires (contained in the DC) and deontic norms (contained in the NAC) for creating new desires. Norm Internalization Bridge Rules depend on the deontic modality of the instance that is being considered.

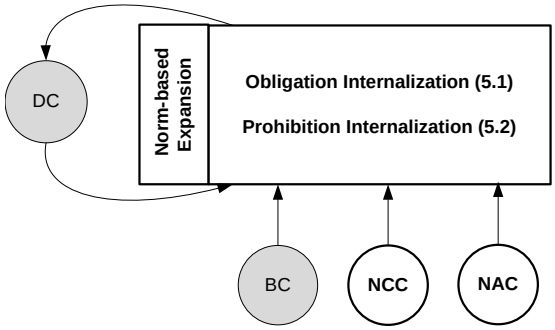

Figure 5.1: Norm-based Expansion for Deontic Norms in the n-BDI Architecture. Contexts are represented as circles, whereas sets of bridge rules that perform similar tasks are represented as boxes in which there are input links, which are the premises of bridge rules, and output links, which represent the conclusions. Gray circles correspond to the basic architecture that has been defined in previous works [CGS11]. The normative extensions are the white elements.

5.2.1 Obligation Internalization

When an agent decides to comply with an obligation, then it internalizes the desire of reaching the state imposed by the obligation. Thus, it creates a positive desire for achieving this obligatory state. Bridge rule for updating the DC with the positive desires derived from obligation instances is defined as follows (see Figure 5.1 Bridge Rule 5.1):

$$\frac{NCC : instance(\langle \mathcal{O}, C', T', self, A', E', S', R' \rangle, \rho_{NCC}), \quad \theta_{will} > \delta_{compliance}}{DC : (\mathcal{D}\ C', f_{internalization}(\rho_{NCC}, \theta_{will}))} \tag{5.1}$$

Any n-BDI agent identifies itself by the $self$ constant. Therefore, those instances that are addressed to the agent itself will be considered for creating new desires. The θ_{will} parameter represents the agent disposition to comply with the instance as a real number within the $[-1, 1]$ interval. If this parameter is equal to 1, then it means that the agent has the highest willingness to comply with the instance. A value equal to 0 means that the agent does not agree to obey the instance. If θ_{will} takes a negative value, then it means that the agent wants to violate the instance deliberately[1]. The concrete definition of θ_{will} is provided in Section 5.3. Once an instance corresponding to an obligation is created, then a new positive desire will be inferred corresponding to the norm condition only if the agent has decided to comply with the norm. To avoid the creation of desires when the willingness to comply with a norm is low, a *norm compliance threshold* has been defined (i.e., $\delta_{compliance} \in [0, 1]$). The definition of this threshold is problem dependent.

Finally, the degree assigned to the normative desire is defined by the $f_{internalization}$ function, which combines the certainty about the activation of the norm (ρ_{NCC}) and the motivation to comply (vs. violate) with norms (i.e., the absolute value of the θ_{will} parameter). Both conditions, the activation of the norm and the motivation to comply with norms, are required for creating a new desire to achieve the obliged condition. Therefore, the combination among the uncertain values that cause the internalization of norms is defined as a symmetric sum[2] [DP85] as follows:

$$f_{internalization}(\rho_{NCC}, \theta_{will}) = \frac{\rho_{NCC} * \theta_{will}}{1 - \rho_{NCC} - \theta_{will} + (2 * \rho_{NCC} * \theta_{will})}$$

5.2.2 Prohibition Internalization

When a prohibition instance is obeyed then a negative desire must be created to represent that the agent does not want to reach the forbidden state. Bridge rule for updating the DC for complying with prohibitions (see Figure 5.1 Bridge Rule 5.2) is defined as:

[1] In this case, a new desire to violate the norm can be created as follows:

$$\frac{NCC : instance(\langle \mathcal{O}, C', T', self, A', E', S', R' \rangle, \rho_{NCC}), \quad \theta_{will} < -\delta_{compliance}}{DC : (\mathcal{D}\ \neg C', f_{internalization}(\rho_{NCC}, \theta_{will}))}$$

[2] Properties of symmetric sums have been described in section 4.4.2.

$$\frac{NCC : instance(\langle \mathcal{F}, C', T', self, A', E', S', R' \rangle, \rho_{NCC}), \quad \theta_{will} > \delta_{compliance}}{DC : (\mathcal{D}\ \neg C', f_{internalization}(\rho_{NCC}, \theta_{will}))} \tag{5.2}$$

Similarly to obligation instances, a prohibition related to a condition C is transformed into a negative desire related to the norm condition.

5.2.3 Permission Internalization

Finally, permission instances do not infer positive or negative desires about the norm condition. In this proposal, we use a closed world assumption where everything is considered as permitted by default. Therefore, permissions define exceptions to the application of more general obligation and prohibition norms. As a consequence, they are only defined for creating an incoherence with these more general norms. For example, in real life there is a general law that forbids drivers to drive faster than the speed limit. However, in case of emergency ambulance drivers are permitted to exceed this limit.

5.3 Determining the Agent Willingness to Norm Compliance

The *assistant* agent needs some procedure to decide what traffic norms will be obeyed or transgressed. The decision procedure of the *assistant* agent is based on the norm-compliance reasoning performed by humans. The *assistant* agent proposes traffic routes to a human user who may execute or not the recommended plan. We consider that the more realistic the agent reasoning is the more reliable the routes are. Thus, the human user will have more confidence on the *assistant* agent if the proposed routes are optimal (or suboptimal) and they seem reasonable. Humans make decisions by balancing their internal motivations (i.e., their own desires) against other external motivations (e.g., social norms or laws). However, each person has his own personality; i.e., each person weights up these factors differently. Next, our human-inspired solution for the n-BDI architecture is described.

The θ_{will} parameter represents the agent willingness to comply with norms. As stated by Conte et al. in [CCD99] "*The decision to comply with a norm is made considering: the value*

of the violation (probability and weight of punishment), the importance of the goal and feelings related to norm violation". Therefore, to calculate this willingness we have mainly considered the works of Elster [Els00, Els89] that analyse factors that sustain norms in human societies. In these works, Elster claims that compliance with norms can be explained by three factors: (i) *self-interest* motivations, which consider the influence of norm compliance and violation on agent's goals; (ii) the *expectations* of being rewarded or sanctioned by others; and (iii) *emotional* factors that are related to internalised emotions such as honour (vs. shame) and hope (vs. fear) that maintain norms. According to this, the θ_{will} parameter is defined as the weighted average among the three *willingness factors* ($\theta_{interest}, \theta_{expectation}$ and $\theta_{emotion}$) as follows:

$$\theta_{will} = \frac{w_{interest} \times \theta_{interest} + w_{expectation} \times \theta_{expectation} + w_{emotion} \times \theta_{emotion}}{w_{interest} + w_{expectation} + w_{emotion}}$$

where the weights $w_{interest}, w_{expectation}$ and $w_{emotion}$ are defined within the $[0, 1]$ interval. The $\theta_{will} \in [-1, 1]$ value is obtained combining the values of the three *willingness factors* ($\theta_{interest}, \theta_{expectation}$ and $\theta_{emotion}$) which are also defined within the $[-1, 1]$ interval. Therefore, we have assumed that the weighted average is a suitable method to derive the central tendency of these three functions.

We consider that norms are sustained by self-interest, enforcement mechanisms and internalised emotions. These three factors determine the agent's will to follow the concrete instance that is being considered. The weights that each agent gives to these factors characterise the agent's personality and do not depend on the instance that is considered. Thus, different types of n-BDI agents can be defined by giving different weights to the *willingness factors*. For example, *egoist* agents [LyLLd06] (i.e., those ones that will accept only norms that benefit their own goals) are defined by defining $w_{interest} = 1$, $w_{expectation} = 0$ and $w_{emotion} = 0$; i.e., by prioritizing their own interests.

Once the intuitive meaning of the *willingness factors* has been provided, their translation in terms of n-BDI agents is explained.

5.3.1 $\theta_{interest}$

This factor evaluates the consequences of a given instance from an utilitarian perspective (i.e., it defines the utility as the good to be maximized). Thus, the $\theta_{interest}$ factor is defined as follows:

$$\theta_{interest} = utility(\langle D, C', T', AgentID, A', E', S', R' \rangle)$$

According to this, the $\theta_{interest}$ factor has been defined by an utility function (*utility*) that computes the utility of an instance. The utility of an instance is defined by considering the direct positive or negative consequence of the norm fulfilment. In case of an obligation, the direct consequence of the norm fulfilment is the norm condition (C'). In case of a prohibition, obeying this prohibition implies that the condition of the norm will be avoided.

Definition 5.3.1 (Utility) *The utility assigned to an instance* $\langle D, C', T', AgentID, A', E', S', R' \rangle$ *is defined as follows:*

$$utility(\langle D, C', T', AgentID, A', E', S', R' \rangle) = \begin{cases} des(C') & \text{if } D = \mathcal{O} \\ des(\neg C') & \text{if } D = \mathcal{F} \end{cases}$$

where des is a function that calculates the desirability of a proposition.

The desirability of a proposition is formally defined as:

Definition 5.3.2 (Desirability) *Given a theory of desires* Γ_{DC}*, the desirability of a proposition* γ *is defined as:*

$$des(\gamma) = \begin{cases} \rho_\gamma - \rho_{\neg\gamma} & \text{if } \Gamma_{DC} \vdash (\mathcal{D}\ \gamma, \rho_\gamma) \text{ and } \Gamma_{DC} \vdash (\mathcal{D}\ \neg\gamma, \rho_{\neg\gamma}) \\ \rho_\gamma & \text{if } \Gamma_{DC} \vdash (\mathcal{D}\ \gamma, \rho_\gamma) \text{ and } \Gamma_{DC} \nvdash (\mathcal{D}\ \neg\gamma, \rho_{\neg\gamma}) \\ -\rho_{\neg\gamma} & \text{if } \Gamma_{DC} \vdash (\mathcal{D}\ \neg\gamma, \rho_{\neg\gamma}) \text{ and } \Gamma_{DC} \nvdash (\mathcal{D}\ \gamma, \rho_\gamma) \\ 0 & \text{otherwise} \end{cases}$$

Therefore, the desirability of a proposition γ (i.e., $des(\gamma)$) is a real value within the $[-1, 1]$ interval such that: the -1 value means that the proposition γ is absolutely rejected, a desirability value of 0 means that the agent is indifferent to γ (i.e., it does not benefit from γ), and 1 means that the agent has maximum preference on γ.

In the proposed case study, the *assistant* agent should make a decision about complying or not with the instance of the Heavy Rain Norm. Let us suppose that the human user has a new and fast car. He likes to show off the power of his new car and he has configured the *assistant* agent with this preference. Since area a_1 is a crowded place, the human user has defined that

he wants to pass across the area a_1 as fast as possible. As a consequence, the *assistant* agent has a desire as the following $(\mathcal{D}\ \neg slow(a_1), 0.9)$. Therefore the interest on obeying this instance is the following:

$$\theta_{interest} =$$
$$utility(\langle \mathcal{O}, slow(a_1), carDriver, self, heavyRain(a_1), \neg heavyRain(a_1), penalty, - \rangle)$$
$$= des(slow(a_1)) = -0.9$$

5.3.2 $\theta_{expectation}$

This factor models the impact of the external enforcement on agents. Specifically, the enforcement mechanism considered in this work consists in a material system of sanctions and rewards that modify the utility that agents obtain when they violate or fulfil norms. According to this, the $\theta_{expectation}$ factor has been defined by an expectation function ($expectedUtility$) that considers how much the agent loses from being penalised and how much it gains from being rewarded. Thus, the $\theta_{expectation}$ factor is defined as follows:

$$\theta_{expectation} = expected_{Utility}(\langle D, C', T', AgentID, A', E', S', R' \rangle)$$

Definition 5.3.3 (Expected Utility) *The expected utility of an instance* $\langle D, C', T', AgentID, A', E', S', R' \rangle$ *is defined by the* $expected_{Utility}$ *function as follows:*

$$expected_{Utility}(\langle D, C', T', AgentID, A', E', S', R' \rangle) =$$
$$\begin{cases} des(R') + des(\neg S') - (des(R') * des(\neg S')) & \text{if } des(R') \geq 0 \text{ and } des(\neg S') \geq 0 \\ des(R') + des(\neg S') + (des(R') * des(\neg S')) & \text{if } des(R') < 0 \text{ and } des(\neg S') < 0 \\ des(R') + des(\neg S') & \text{otherwise} \end{cases}$$

where des is defined as before.

Since the fulfilment of the norm implies that the agent will be both rewarded and not sanctioned, the expected utility is defined as the combination of the desirability of R and $\neg S$. Again, we

have considered the MYCIN rules [SB75] for combining the desirability of the two consequences of norm fulfilment. MYCIN rules are a variable fusion operator that behaves as follows:

- if both $des(R)$ and $des(\neg S)$ are positive, the $expected_{Utility}$ function provides a combined desirability value higher than each individual factor ($des(R)$ and $des(\neg S)$);
- if both $des(R)$ and $des(\neg S)$ are negative, the $expected_{Utility}$ function results in a stronger undesirability than each individual factor;
- otherwise the $expected_{Utility}$ function results in a combined desirability that is a compromise among the two desirability values.

For simplicity it has been assumed that there is a perfect external enforcement that always punishes offenders and rewards obedience. However, if agents are able to perceive the probability of being punished or rewarded, then the desirability of sanctions and rewards can be pondered with their observed probabilities. The determination of the probability of being punished and rewarded is beyond the scope of this thesis. However, this information may be inferred in the n-BDI proposal by observing the number of times that a norm is rewarded or sanctioned.

In the proposed case study, let us suppose that the human user is a rich man who does not care about money. Therefore, he is not very worried about paying penalties. Thus, the *assistant* agent has a desire as the following $(\mathcal{D}\ \neg penality, 0.25)$ and the enforcement of this norm is not very relevant to the agent:

$$\theta_{expectation} =$$
$$expected_{Utility}(\langle \mathcal{O}, slow(a_1), carDriver, self, heavyRain(a_1), \neg heavyRain(a_1), penalty, - \rangle) =$$
$$des(-) + des(\neg penalty) = 0 + 0.25 - (0 * 0.25) = 0.25$$

5.3.3 $\theta_{emotion}$

This factor models the emotions triggered by the social evaluation of the agent's behaviour. Thus, the $\theta_{emotion}$ factor models the social cost of violating norms, whereas the $\theta_{expectation}$ factor models the economic cost. The term emotion is used in this work for representing the valued reaction of agents (i.e., the agent's cognitive interpretation) with respect to some aspect of

the world (i.e., the reality) [OCC88]. n-BDI agents do not have an explicit representation and reasoning about emotions as in occurs in other proposals such as [DTM09, SDM07]. In fact, our proposal is not to build emotional agents, but to develop norm-autonomous agents capable of understanding the most relevant emotions that are involved in the decision about norm compliance. Specifically, n-BDI agents are capable of anticipating, exhibiting and explaining those human emotions that are involved with the normative decisions. Thereby, the decisions about norm compliance are also based on other criteria beyond utility.

As argued by Elster in [Els89, Els00], in humans the behaviour guided by norms is sustained by the desire to avoid the disapproval of others. Following Elster's proposal, when the violation of norms is greeted with condemnation, then self-*attribution* emotions (i.e., shame) are triggered on the offender. Moreover, the situations that are predicted to occur when norms are violated may cause *prospect* emotions (i.e., hope and fear) on the offender. According to this, the $\theta_{emotion}$ factor has been implemented by a function ($anticipated_{emotions}$) that anticipates the emotions that will be triggered if a given instance is violated:

$$\theta_{emotion} = anticipated_{emotions}(\langle D, C', T', AgentID, A', E', S', R' \rangle)$$

Thus, in the n-BDI architecture there are two types of emotions by which an agent decides to comply or not with norms: self-attribution emotions (e_a), which calculate the disapproving of one's own censurable action; and prospect emotions (e_p), which calculate the fear (vs. hope) about the prospect of undesirable (vs. desirable) events. The self-attribution emotions (e_a) are represented as a real value within the $[0, 1]$ interval that determines the evaluation (i.e., attribution) that the agent makes about itself if it violates the norm. Therefore, e_a sustains norm obedience. Prospect emotions (e_p) can sustain either the obedience or violation of norms; e.g., in some conditions the violation of norms may entail desirable consequences. Thus, e_p is a real value within the $[-1, 1]$ interval that considers the possible outcomes of violating an instance. Positive values for e_p mean that the agent fears to violate the instance, since it beliefs that the violation may entail undesirable consequences. On the contrary, a negative value means that the agent considers norm violation as a hopeful possibility, since it would entail desirable consequences. The degrees of these two emotions (e_a and e_p) are combined by the $anticipated_{emotions}$ function, which has been defined considering the MYCIN[3] [SB75] rules for

[3]The properties of the MYCIN rules have been described previously in this section.

combining two pieces of information supporting the same event as follows:

Definition 5.3.4 (Anticipated Emotions) *Given an instance* $\langle D, C', T', AgentID, A', E', S', R' \rangle$ *the value of the anticipated emotions that will be triggered if an agent violates this instance is defined as:*

$$anticipated_{emotions}(\langle D, C', T', AgentID, A', E', S', R' \rangle) = \begin{cases} e_a + e_p - (e_a * e_p) & \text{if } e_p > 0 \\ e_a + e_p & \text{otherwise} \end{cases}$$

where e_a *is the value of the self-attribution emotions which are calculated by the* $f_{attribution}$ *function (see Definition 5.3.6) and* e_p *is the value of the prospected emotions calculated by the* $f_{prospect}$ *function (see Definition 5.3.7).*

Thus, the $anticipated_{emotions}$ function calculates the agent emotional disposition to comply with an instance as a real number within the $[-1, 1]$ interval. For example, a -1 value means that the agent feels that it does not want to follow the norm.

In order to allow n-BDI agents to estimate the value of these two emotions (e_a and e_p) an emotional model susceptible to be implemented in a software agent is required. One of the emotional models that have made a deeper impact on the MAS field is the one developed by *Ortony, Clore and Collins* (OCC) in [OCC88]. This work proposes a taxonomy of emotions according to their eliciting conditions. The representation of mental and normative elements in the n-BDI architecture fits perfectly the cognitive factors considered by the OCC model as determinant for establishing the type and intensity of the emotions that are involved in the norm-reasoning. Therefore, the OCC model has been considered as a reference for anticipating the emotions triggered by a given instance. Next, the implementation of each one of these two emotional functions (i.e., $f_{attribution}$ and $f_{prospect}$) in the n-BDI architecture is explained:

- *Self-Attribution Emotions.* According to the OCC model, shame is a self-attribution emotion that is elicited by the actions that have been performed by the agent itself. Specifically, when humans evaluate the actions that themselves do, this evaluation is made with respect to norms. Therefore, actions of agents are self-evaluated as censurable insofar as these actions contradict the norms. In this case, the *praiseworthiness* of these actions is the most relevant factor in the intensity of attribution emotions. In particular, the shame that the agent will feel if it violates a given instance is defined by considering

the importance (i.e., the salience) of these norms that are generalizations of this instance[4]. The set of norms that are generalizations of a given instance is formally defined as follows:

Definition 5.3.5 (Instance Generalization) *Given a belief theory Γ_{BC}, a normative theory Γ_{NAC} and an instance $\langle D, C', T', AgentID, A', E', S', R' \rangle$ that has been created out of some norm contained in Γ_{NAC}; the set of norms that are a generalization of this instance is defined as follows:*

$$generalization(\langle D, C', T', AgentID, A', E', S', R' \rangle) =$$
$$\{norm(\langle D_i, C_i, T_i, A_i, E_i, S_i, R_i \rangle, \rho_i) \in \Gamma_{NAC} | D_i = D, \Gamma_{BC} \vdash play(self, T_i)$$
$$\text{and exists a substitution } \sigma_i \text{ such that } C' \vdash \sigma_i(C_i), A' \vdash \sigma_i(A_i) \text{ and } E' \vdash \sigma_i(E_i)\}$$

Therefore, any norm can be seen as a generalization of a given instance if the two normative propositions have the same deontic modality, the norm is addressed to some of the roles that are played by the agent, and there is a substitution such that the norm can be derived from the instance. According to this definition, the $f_{attribution}$ is defined as the average among the salience values of these norms that are a generalization of a given instance as follows:

Definition 5.3.6 (Self-Attribution Emotions) *Given an instance i, the intensity of the self-attribution emotions triggered by the violation of the instance is defined by the $f_{attribution}$ function as follows:*

$$f_{attribution}(i) = \rho_{max}$$

where ρ_{max} is a real value within the interval $[0, 1]$ such that $\exists norm(n_{max}, \rho_{max}) \in generalization(i)$ and $\forall norm(n_i, \rho_i) \in generalization(i) : \rho_{max} \geq \rho_i$.

Thus, the intensity of the shame emotion that will be elicited if an instance is violated is defined as the salience of the most important (i.e., salient) norm that is a generalization of this particular instance.

[4]Each instance is created out of a single norm. However, an instance can be seen as a particularization (i.e., instantiation) of more than one norm.

- *Prospect Emotions.* According to the OCC model, the hope (vs. fear) emotion is triggered when a desirable (vs. undesirable) event is predicted. Therefore, the main factors on the intensity of hope (vs. fear) are the probability of the predicted event and the desirability (vs. undesirability) of this event. The fear and hope emotions that may be triggered if an instance is violated are defined by considering the desirability and probability of the consequences of the violation as follows:

Definition 5.3.7 (Prospect Emotions) *Given a theory of beliefs Γ_{BC} and an instance $\langle D, C', T', AgentID, A', E', S', R' \rangle$ the prospect emotions triggered by the violation of this instance is defined by the $f_{prospect}$ function as follows:*

$$f_{prospect}(\langle D, C', T', AgentID, A', E', S', R' \rangle) = \begin{cases} \dfrac{-\sum_{i=1}^{n} \beta_i * des(\gamma_i)}{\sum_{i=1}^{n} \beta_i} & \text{if } D = \mathcal{O} \text{ and there is a set of } n \text{ beliefs } \{..., (\mathcal{B}\ \neg C' \rightarrow \gamma_i, \beta_i), ...\} \text{ where each belief } (\mathcal{B}\ \neg C' \rightarrow \gamma_i, \beta_i) \in \Gamma_{BC} \\ \dfrac{-\sum_{i=1}^{n} \beta_i * des(\gamma_i)}{\sum_{i=1}^{n} \beta_i} & \text{if } D = \mathcal{F} \text{ and there is a set of } n \text{ beliefs } \{..., (\mathcal{B}\ C' \rightarrow \gamma_i, \beta_i), ...\} \text{ where each belief } (\mathcal{B}\ C' \rightarrow \gamma_i, \beta_i) \in \Gamma_{BC} \end{cases}$$

Thus, $f_{prospect}$ is a function that calculates the prospect emotions triggered by the violation of the instance as a real value within the $[-1, 1]$ interval. A positive value sustains compliance with the instance. Specifically, it means that the violation of the norm raises the agent's fears. A negative value of the $f_{prospect}$ function sustains the violation of norms. It occurs when the agent hopes that the violation of the norm entails desirable consequences. Therefore, in case of an obligation instance the prospect emotions triggered by the violation of the instance are defined as the mean among the desirability of the effects of the violation of the obligation (i.e. the negation of the norm condition ($\neg C$)). In case

of a prohibition instance, its violation entails the achievement of the norm condition (C). In both cases, the desirability of the consequences of the violation has been weighted by the probability of their occurrence (β_i). In accordance with the previous definitions of the *willingness factors*, which define positive values as compliance sustaining, the $f_{prospect}$ function has been defined as minus the weighted mean of the desirability of the effects of the violation.

In the proposed case study, the value of the attribution emotion calculated by the *assistant* agent is 0.35, which is the salience of the Heavy Rain Norm:

$$e_a = f_{attribution}(\mathcal{O}, slow(a_1), carDriver, self, heavyRain(a_1), \neg heavyRain(a_1), penalty, -\rangle) = 0.35$$

The *assistant* agent calculates the value of the prospect emotion by considering the consequences of not reducing the speed. Specifically, the *assistant* agent considers that not reducing the speed may cause an accident with a probability of 25% —i.e., the *assistant* agent has a belief such as $(\mathcal{B}\ \neg slow \rightarrow accident, 0.25)$—. The human user does not want to cause an accident —i.e., the *assistant* agent has a desire such as $(\mathcal{D}\ \neg accident, 1)$—. Therefore the value of the prospect emotion is 1:

$$e_p = f_{prospect}(\langle \mathcal{O}, slow(a_1), carDriver, self, heavyRain(a_1), \neg heavyRain(a_1), penalty, -\rangle) = \frac{-(0.25*-1)}{0.25} = 1$$

and the value of the anticipated emotions is 1:

$$\theta_{emotion} = anticipated_{emotions}(\mathcal{O}, slow(a_1), carDriver, self, heavyRain(a_1), \neg heavyRain(a_1), penalty, -\rangle) = e_a + e_p - (e_a * e_p) = 0.35 + 1 - (0.35 * 1) = 1$$

Let us assume that the human user has configured the *assistant* agent to consider the three willingness factors equally. Therefore willingness of the *assistant* agent to comply with the instance is 0.12:

$$\theta_{will} = \frac{1 \times \overbrace{-0.9}^{\theta_{interest}} + 1 \times \overbrace{0.25}^{\theta_{expectation}} + 1 \times \overbrace{1}^{\theta_{emotion}}}{3} = 0.12$$

Remember that $\theta_{interest}$ and $\theta_{expectations}$ have been calculated previously in Sections 5.3.1 and

5.3.2, respectively. The *assistant* agent decides to comply with the instance (the creation of this instance has been explained in Section 4.4.2.1). Therefore, it creates a new desire to achieve the norm condition according to the Obligation Internalization Bridge Rule (see Equation 5.1) as follows:

$$\frac{NCC : instance(\langle \mathcal{O}, slow(a_1), carDriver, self, heavyRain(a_1), \neg heavyRain(a_1), penalty, - \rangle, 0.25), \quad 0.12 > 0.1}{DC : (\mathcal{D}\ slow(a_1), f_{internalization}(0.75, 0.12))}$$

where the compliance threshold ($\delta_{compliance}$) is set to 0.1. Thus, the degree of the new desire is 0.29:

$$f_{internalization}(0.75, 0.12) = \frac{0.75 * 0.12}{1 - 0.75 - 0.12 + (2 * 0.75 * 0.12)} = 0.29$$

and thus the DC contains new a proposition such as:

$$(\mathcal{D}\ slow(a_1), 0.29)$$

5.4 Experimental Results

This section illustrates the performance of the different agent types with respect to their decisions about norm compliance, which are modelled using the willingness function. Therefore, other problems that have been faced by chapters, such as norm conflicts (see Chapter 7), have been omitted.

5.4.1 Simulation Description

We considered a scenario with the parameters that we sum up in Table 5.1. As previously mentioned, the goal of this simulation is to illustrate the behaviour of the main types of agents. Specifically, 7 different agent personalities have been compared. Therefore, in the simulation one agent of each type is created. These agents are affected by the same set of norms and instances. Moreover, all agents have the same desires and beliefs. Therefore, the only difference among agents is the way in which they make decisions about norm compliance; i.e., how they decide about which instances will be obeyed and which ones will be violated.

Parameter	Value
# of agents	7
# of goals	[0,100]
# of explanatory relationships	[0,100]
Norm compliance threshold ($\delta_{compliance}$)	0.1
# of norms	100
# of instances	500
Norm acceptance degree (ρ_{NAC})	[0,1]
Norm relevance degree (ρ_{NCC})	[0,1]
# of simulations	100

Table 5.1: Parameters used in the simulations

5.4.1.1 Agent Definition

Agents pursue a set of desirable states or *goals* that are randomly generated. Each goal is a tuple $\langle g_i, v_i, r_i \rangle$, where $g_i \in \mathcal{L}$ is the logic proposition that represents the desired state, $v_i \in [0.75, 1]$ is the desirability degree, and $r_i \in [0, 5]$ is a real value that represents the similarity between g_i and the least similar proposition that is also desired (i.e., if a proposition is desired with a certain degree then it makes sense that similar propositions are also desired with a lower degree). The more similar a given state and a desired state are, the more desirable the state is[5]. The size of the goal set is randomly defined in each execution within the $[0, 100]$ interval. Goals are also randomly generated: each desired proposition g_i is defined as a random proposition in $\mathcal{L}$; v_i and r_i also take random values. Figure 5.2 shows an example of the desire distribution for an agent. In this graph we use a bijective function π that associates a real value within the interval $[-50, 50]$ to each proposition $\gamma \in \mathcal{L}$. The more similar two propositions γ_1 and γ_2 of $\mathcal{L}$ are, the closer the values $\pi(\gamma_1)$ and $\pi(\gamma_2)$ are. Moreover, for all γ in $\mathcal{L}$: $\pi(\neg\gamma) = -\pi(\gamma)$. Thus, in Figure 5.2 the X-axis illustrates the real value that corresponds to each logic proposition[6], whereas the Y-axis shows the desirability degree of this proposition[7]. The desirability distribution shown is the maximum among the desirabilities of propositions in $\mathcal{L}$ with respect to the goals.

Besides the desirability of propositions, agents also use explanatory relationships among propositions for making decisions about norm compliance. These explanatory relationships are

[5]Given a goal $\langle g_i, v_i, r_i \rangle$, the desirability degree of a proposition $\gamma \in \mathcal{L}$ with respect to this goal is calculated as follows:

$$\begin{cases} \frac{(\pi(\gamma)-\pi(g_i)+r_i)v_i}{r_i} & \text{if } \pi(g_i) - r_i < \pi(\gamma) <= \pi(g_i) \\ \frac{(\pi(g_i)+r_i-\pi(\gamma))v_i}{r_i} & \text{if } \pi(g_i) < \pi(\gamma) < \pi(g_i) + r_i \\ 0 & \text{otherwise} \end{cases}$$

[6]$\{\pi(\gamma) : \gamma \in \mathcal{L}\}$.

[7]$\{\rho_\gamma : \Gamma_{DC} \vdash (\mathcal{D}\ \gamma, \rho_\gamma)$ and $\gamma \in \mathcal{L}\}$.

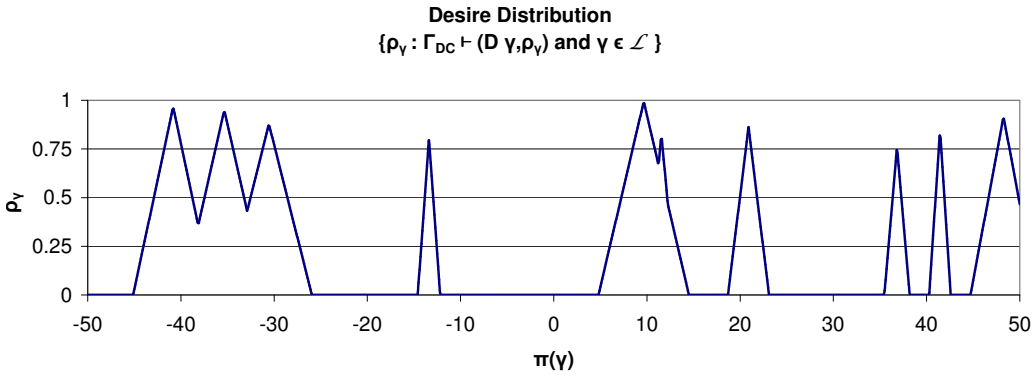

Figure 5.2: Desire distribution of a randomly generated agent. The X-axis represents the real value that corresponds to each proposition $\gamma \in \mathcal{L}$ and the Y-axis shows the desirability degree.

represented as graded beliefs such as $(\mathcal{B}\ \alpha \rightarrow \gamma, \beta)$, which means that α explains γ with a probability of β. For this experimentation, these relationships are randomly generated. The antecedent (α) and consequence (γ) of an explanatory relationship are random propositions of $\mathcal{L}$. The probability of these relationships (β) is a random real within $[0, 1]$. For example, Figure 5.3 illustrates a bubble chart that contains 100 explanatory relationships that have been randomly generated. In each execution, agents know a random number of explanatory relationships that ranges within the $[0, 100]$ interval.

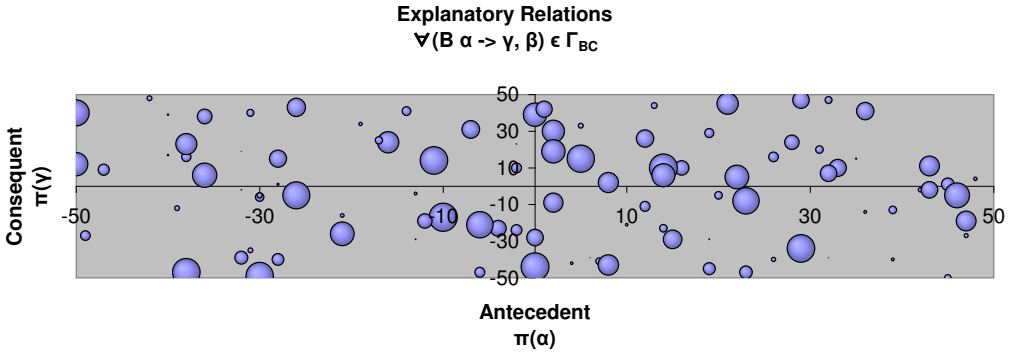

Figure 5.3: Explanatory relationship graph. Both X and Y axis represent the real value $\pi(\gamma)$ that corresponds to each proposition $\gamma \in \mathcal{L}$. Specifically, X-axis represents the antecedent of an explanatory relationship and the Y-axis represents the consequent. The area of the plots depends on the probability assigned to each explanatory relationship.

5.4.1.2 Norm Definition

In each execution 100 deontic norms are randomly generated. Specifically, the norm condition (C), the activation (A) and expiration (E) conditions, and the sanction (S) and reward (R) of

each norm are randomly defined as propositions of $\mathcal{L}$ (i.e., $\pi(C)$, $\pi(A)$, $\pi(E)$, $\pi(S)$ and $\pi(R)$ are real values within the $[-50, 50]$ interval). The norm acceptance degree (ρ_{NAC}) gets a random value within the interval $[0, 1]$.

From each norm 5 instances are randomly created (i.e., a total amount of 500 instances are created in each execution). There must be some similarities between a norm and the instances that are created out of this norm. Thus, the instantiation of the norm condition, the activation and expiration conditions and the sanction and reward are propositions C', A', E', S', R' in $\mathcal{L}$ such that $\pi(C'), \pi(A'), \pi(E'), \pi(S'), \pi(R')$ take their values randomly within the intervals $(\pi(C) - 1, \pi(C) + 1)$, $(\pi(A) - 1, \pi(A) + 1)$, $(\pi(E) - 1, \pi(E) + 1)$, $(\pi(S) - 1, \pi(S) + 1)$, $(\pi(R) - 1, \pi(R) + 1)$, respectively. The main purpose of this simulation is to compare the performance of the different agent personalities with respect to the norm compliance decision. This decision is not affected by the relevance of instances, but only by θ_{will} and $\delta_{compliance}$. Therefore, the value of ρ_{NCC} is assigned a random value within the $[0, 1]$ interval.

5.4.1.3 Agent Types

In the n-BDI architecture the decisions about norm compliance are made by considering three different factors: self-interest ($\theta_{interest}$), the enforcement mechanisms ($\theta_{expectation}$) and the emotions triggered by the violation of norms ($\theta_{emotion}$). These three factors are combined in a single value (θ_{will}) that is defined as a weighted average among these three willingness factors. Therefore, different agent personalities can be modelled according to the definition of the weights $w_{interest}$, $w_{expectation}$ and $w_{emotion}$. The three basic personalities are: *egoist*, *cautious* and *emotional*:

- *Egoist agents* ($w_{interest} = 1$, $w_{expectation} = 0$ and $w_{emotion} = 0$) only follow those norms that favour their goals or that avoid some undesirable state. For example, in case of obligation instances egoist agents only consider the desirability of the norm condition ($des(C')$) for deciding about norm compliance. In case of prohibition instances egoist agents only consider the desirability of the negation of the norm condition ($des(\neg C')$) that will be avoided if the instance is fulfilled.

- *Cautious agents* ($w_{interest} = 0$, $w_{expectation} = 1$ and $w_{emotion} = 0$) comply with norms when they want to avoid the sanctions or when they are interested on the rewards. Thus, the values obtained by the willingness function depend on the values of both $des(\neg S')$ and

$des(R')$.

- *Emotional agents* ($w_{interest} = 0$, $w_{expectation} = 0$ and $w_{emotion} = 1$) only consider the emotions that will be elicited if norms are violated. As explained in Section 5.3, n-BDI agents are capable of anticipating both attribution and prospect emotions:

  - *Attribution emotion.* As explained before, the $f_{attribution}$ function is defined as the maximum among the acceptance values of those norms that are a generalization of a given instance. According to the formal definition of generalization, any deontic norm ($\langle D, C, T, A, E, S, R\rangle$) can be seen as a generalization of a given deontic instance ($\langle D, C', T', AgentID, A', E', S', R'\rangle$) if the two normative propositions have the same deontic modality and there is a substitution such that the norm can be derived from the instance (i.e., $\exists\sigma : C' \vdash \sigma(C) \wedge A' \vdash \sigma(A) \wedge E' \vdash \sigma(E)$).

    In the simulations we have considered that there is a substitution σ such as to propositions γ and γ' satisfy $\gamma' \vdash \sigma(\gamma)$ when $\pi(\gamma') \in (\pi(\gamma)-1, \pi(\gamma)+1)$[8]. Therefore, an instance ($\langle D, C', T', self, A', E', S', R'\rangle$) is a generalization of a norm ($\langle D, C, T, A, E, S, R\rangle$) when $\pi(C') \in (\pi(C) - 1, \pi(C) + 1)$, $\pi(E') \in (\pi(E) - 1, \pi(E) + 1)$ and $\pi(A') \in (\pi(A) - 1, \pi(A) + 1)$.

  - *Prospect emotion.* The main factor on the intensity of prospect emotion is the probability of the consequences of norm violation and the desirability (vs. undesirability) of these consequences.

    The consequences of violating an instance are calculated by considering the explanatory relationships that an agent knows. For example the consequences of violating an instance such that $\langle \mathcal{O}, C', T', self, A', E', S', R'\rangle$ are calculated by considering those explanatory relationships that have as antecedent a proposition α such that $int(\pi(\alpha)) = int(-\pi(C'))$[9].

[8] According to the way in which instances are generated, the instantiation of the a proposition $\gamma \in \mathcal{L}$ is defined as a proposition γ' in $\mathcal{L}$ such that $\pi(\gamma')$ is a real value within the $(\pi(\gamma) - 1, \pi(\gamma) + 1)$ interval.

[9] $int(x) = \begin{cases} \lfloor x \rfloor & \text{if } x \geq 1 \\ 0 & \text{if } -1 < x < 1 \\ \lceil x \rceil & \text{if } x \leq -1 \end{cases}$

5.4.2 Results

Figure 5.4 illustrates the performance of the different types of agents with respect to their decisions about norm compliance. This decision is modelled by the θ_{will} parameter. Specifically, in Figure 5.4 each agent type has been labelled according to the values given to the weights $w_{interest}$, $w_{expectation}$ and $w_{emotion}$. The values obtained by the θ_{will} function have been classified again in three categories according to the values of the *norm compliance threshold* ($\delta_{compliance}$): deciding to *violate* (i.e., when θ_{will} ranges within the $[-1, -\delta_{compliance})$ interval); deciding to *ignore* (i.e., when θ_{will} ranges within the $[-\delta_{compliance}, \delta_{compliance}]$ interval); and deciding to *obey* (i.e., when θ_{will} ranges within the $(\delta_{compliance}, 1]$ interval). Deciding to violate an instance means that the agent will try to behave contrary to the pattern of behaviour specified by the instance. Deciding to obey an instance means that the agent will try to follow the pattern of behaviour specified by the instance. Deciding to ignore an instance means that the agent will not change its behaviour regardless of the instance. Thus, the instance would be either obeyed or violated. Specifically, Figure 5.4 shows the percentage of instances that belong to each one of the willing categories (i.e. *violate*, *ignore* and *obey*) when $\delta_{compliance}$ is set to 0.1. This simulation has been repeated 100 executions to support findings.

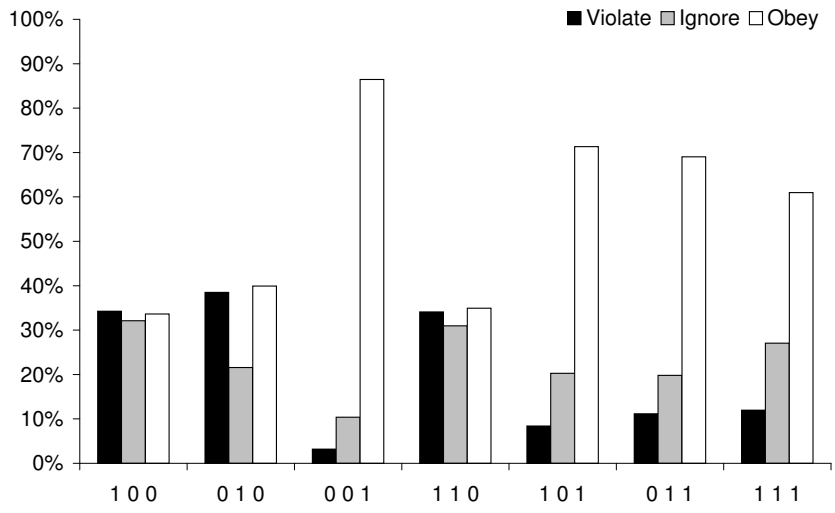

Figure 5.4: Percentage of instances that belong to each willingness category on average when $\delta_{compliance}$ takes value 0.1. For each number in the X-axis, the first value stands for $w_{interest}$, the second value stands for $w_{expectation}$, and the last value stands for $w_{emotion}$. Thus, "1 0 0" represents egoist agent, "0 1 0" represent cautious agent, and "0 0 1" represent egoist agent.

Regarding the three main agent personalities, it can be concluded that egoist agents (labelled as 1 0 0) are the most prone to ignore norms, since they only consider if the norm condition favours or hinders their goals. Cautious agents (labelled as 0 1 0) are not as prone to ignore

norms, i.e., the percentage of ignored instances is lower. This can be explained by the fact that cautious agents consider whether either the reward or the negation of the sanction favour their goals. Therefore, the percentage of instances that are indifferent in cautious agents is lower. In case of egoist and cautious agents there is a symmetric distribution of instances in the three willingness categories; i.e., egoist and cautious agents decide to obey as many norms as they decide to violate. This is explained by the fact that norms and desires are randomly generated and, as a consequence, norms favour or hinder the agent goals with the same probability. Finally, emotional agents (labelled as 0 0 1) are the most willing to obey norms; i.e., they are the most norm-oriented. This is explained by the fact that the attribution emotion (modelled by the $f_{attribution}$ function) only sustains norm obedience. Moreover, the percentage of ignored norms in emotional agents is the lowest. This is explained by the combination among the prospect (modelled by the $f_{prospect}$ function) and the attribution emotion. The prospect emotion considers the desirability of all the possible consequences of violating an instance. Thus, it is possible that the negative effects counteract the positive ones and the values obtained by the $f_{prospect}$ function are near to 0. This value is combined with the value calculated by the $f_{attribution}$ function, which is always positive, and θ_{will} takes a value higher than $\delta_{compliance}$.

Other agent personalities can be defined from these three basic personalities by giving different values to the weights $w_{interest}$, $w_{expectation}$ and $w_{emotion}$. In this simulation, we have also analysed the behaviours of agents that use a mixed strategy for making decisions about norm compliance. Therefore, two or more willingness factors are considered in the calculation of the θ_{will} parameter[10]. As expected (see Figure 5.4), all agents that consider emotions, (i.e., $w_{emotion} = 1$) have a tendency to decide to obey norms. Specifically, agents that consider the three willingness factors (i.e., $w_{interest} = 1$, $w_{expectation} = 1$ and $w_{emotion} = 1$), labelled as 1 1 1, comply with less norms than the rest of emotional agents, labelled as 1 0 1 and as 0 1 1; since the influence of emotions is reduced by the other two factors. In case of agents that consider interest and expectation (i.e., $w_{interest} = 1$, $w_{expectation} = 1$ and $w_{emotion} = 0$), the percentage of instances that are ignored is higher than in cautious agents. This is explained by the fact that the norm conditions, the sanctions, and the rewards are randomly generated; i.e., there is not any relationship among a norm and its enforcement. Therefore, it is possible that a norm favours one of the agent goals but the reward that the agent will receive hinders another goal. In this situation, the agent has motivations for violating the norm and also motivations for

[10]For simplicity we have only considered these agent types in which the $w_{interest}, w_{expectation}, w_{emotion} \in \{0, 1\}$

following it. Thus, it decides to ignore the norm. Also due to the random generation of norms and desires, these agents decide to violate as many norms as they decide to obey.

5.4.2.1 Compliance Threshold $\delta_{compliance}$

The previous simulation has been repeated assigning different values to the compliance threshold ($\delta_{compliance}$). Figures 5.5 and 5.6 show the percentage of instances that belong to each one of the willing categories when $\delta_{compliance}$ is 0.05 and 0.2, respectively. As expected, the lower value takes the compliance threshold the lower instances are ignored. On the contrary, when the compliance threshold takes higher values the percentage of ignored instances increases. However, similar relationships among the agent personalities can be observed. The most relevant difference among the results shown by these figures is the relationship among agents that consider interest and expectation (i.e., $w_{interest} = 1$, $w_{expectation} = 1$ and $w_{emotion} = 0$), egoist agents and cautious agents. As Figure 5.6 shows when $\delta_{compliance}$ is 0.2 agents that consider interest and expectation (labelled as 1 1 0) ignore more norms than egoist agents (labelled as 1 0 0) and cautious agents (labelled as 0 1 0). This is explained by the fact that the two norm compliance factors have been combined as an arithmetic mean (i.e., as a weighted mean where $w_{interest} = 1$ and $w_{expectation} = 1$). The arithmetic mean always behaves as a compromise operator and, as a consequence, $min(\theta_{interest}, \theta_{expectation}) \leq \theta_{will} \leq max(\theta_{interest}, \theta_{expectation})$. Therefore, norms are obeyed (vs. violated) only when both $\theta_{interest}$ and $\theta_{expectation}$ are higher (vs. lower) than $\delta_{compliance}$ (vs. $-\delta_{compliance}$). As mentioned above, the norm conditions, the sanctions and the rewards are randomly generated in an independent way, which makes difficult that both $\theta_{interest}$ and $\theta_{expectation}$ take values higher (vs. lower) than $\delta_{compliance}$ (vs. $-\delta_{compliance}$). In fact, the percentage of ignored norms increases more in all agents that combine two or more compliance factors.

5.4.2.2 Acceptance of Norms

With the aim of determining the effect of the acceptance of norms on the decisions about norm compliance, we also run out simulations varying the acceptance degree of norms. Figure 5.7 shows the results obtained when the acceptance of norms is very low; i.e. $\rho_{NAC} \in [0, 0.25]$. Similarly, Figures 5.8, 5.9 and 5.10 show the results obtained when the acceptance degrees range within the $[0.25, 0.5]$, $[0.5, 0.75]$ and $[0.75, 1]$ intervals, respectively. As one could expect from the definitions of the willingness functions only those agents that consider emotions are

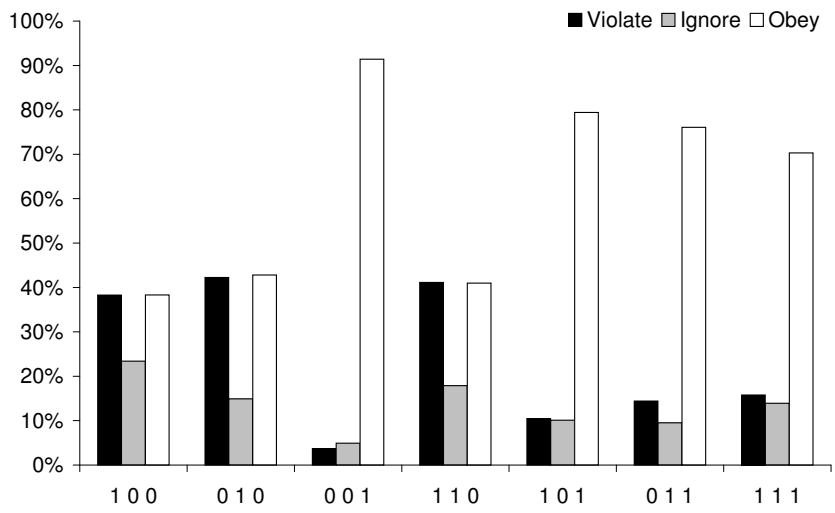

Figure 5.5: Percentage of instances that belong to each willingness category on average when $\delta_{compliance}$ takes value 0.05

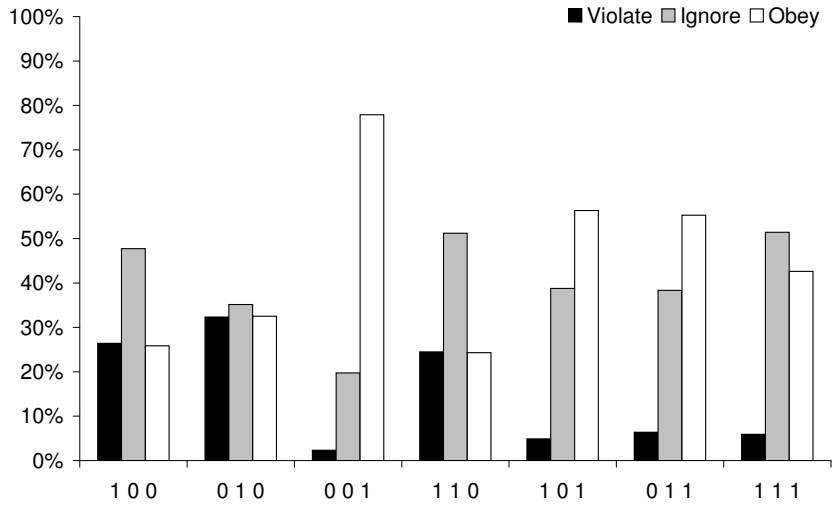

Figure 5.6: Percentage of instances that belong to each willingness category on average when $\delta_{compliance}$ takes value 0.2

affected by the acceptance of norms. When the acceptance of norms is very low (see Figure 5.7) the percentage of obeyed norms in all agents that consider emotions decreases. As the acceptance of norms increases (see Figure 5.8) the percentage of obeyed norms increases. In case of emotional agents ($w_{interest} = 0$, $w_{expectation} = 0$ and $w_{emotion} = 1$) the percentage of obeyed norms is higher than the average results. In case of agents that consider emotions and other factors the influence of the acceptance of norms is reduced by the other factors. As a consequence, these agents still obey less norms that in the average results. When the acceptance of norms is high or very high (see Figures 5.9 and 5.10) all emotional agents are highly influenced by the acceptance values and they obey more norms than in the average

results. In summary, in situations where the acceptance of norms is high it is more suitable to not use emotional agents, since they would behave as norm-oriented agents that follow almost all norms.

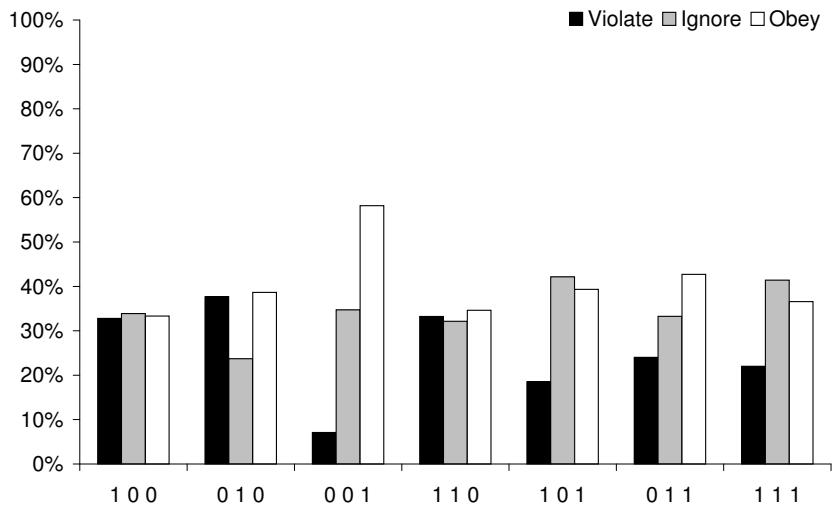

Figure 5.7: Percentage of instances that belong to each willingness category on average when when $\rho_{NAC} \in [0.0, 0.25]$

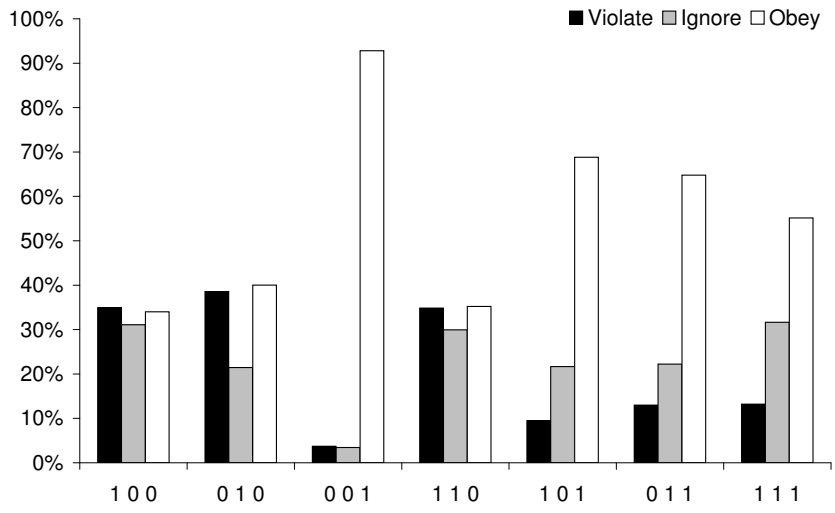

Figure 5.8: Percentage of instances that belong to each willingness category on average when $\rho_{NAC} \in [0.25, 0.5]$

5.4.2.3 Agent Goals

In this simulation, we run out simulations varying the number of goals that an agent pursues. Figures 5.11, 5.12 and 5.13 show the results obtained when the number of goals is 10, 50 and

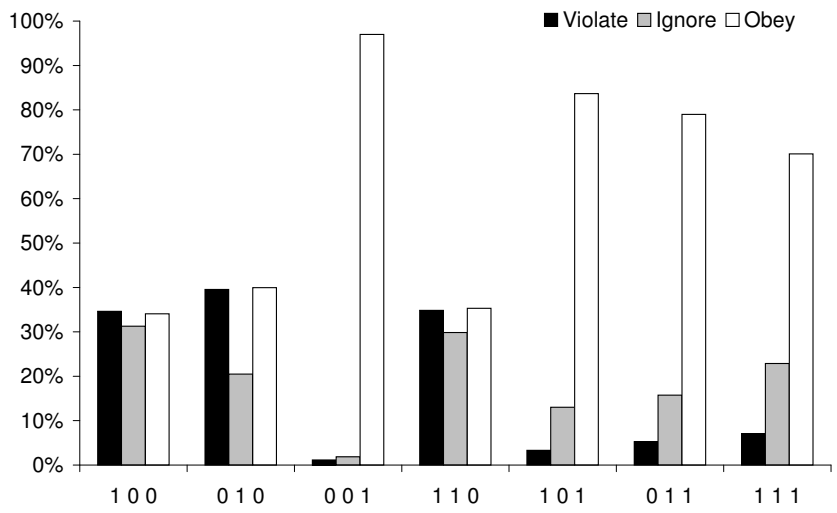

Figure 5.9: Percentage of instances that belong to each willingness category on average when $\rho_{NAC} \in [0.5, 0.75]$

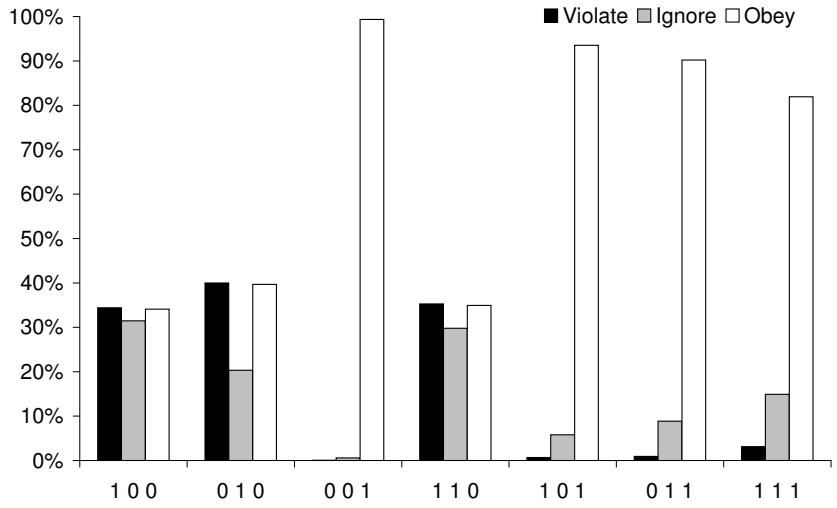

Figure 5.10: Percentage of instances that belong to each willingness category on average when $\rho_{NAC} \in [0.75, 0.1]$

100, respectively. In light of these results, we can conclude that agents that consider self-interest and expectation factors are the most affected by the number of goals.

In case of egoist agents, when the number of goals is low (see Figure 5.11) agents have very few information for making decisions about norm compliance and the percentage of ignored norms increases. As the number of goals increases (see Figure 5.12), the percentage of ignored norms decreases. However, when there is a high number of goals (see Figure 5.13), there is a high probability that a proposition and its negation are simultaneously desired. In this situation agents cannot conclude if norms hinder or favour its goals so they decide to ignore norms.

Cautious agents (labelled as 0 1 0)) consider the desirability of two different propositions

(i.e., the reward and the negation of the sanction). For this reason, even when the number of goals is low (see Figure 5.11), cautious agents have enough information for making decisions about norm compliance and the percentage of ignored norms is lower than the average results. Again as the number of goals increases (see Figure 5.12), the number of ignored norms decreases. Finally, when there is a high number of goals (see Figure 5.13) the percentage of ignored norms on cautions agents is lightly higher than the average results.

Finally, those agents that take into account both the self-interest and the expectation factors ($w_{interest} = 1$, $w_{expectation} = 1$ and $w_{expectation} = 0$) are less affected by the number of goals and the percentage of ignored norms is more similar to the average results. Only when the number of goals is high (see Figure 5.13), the percentage of ignored norms is higher than the average results. This is due to the fact that these agents combine two factors that are not conclusive in these circumstances (when the number of goals is high).

In summary, when the number of goals that an agent pursues is low it is better not to use pure egoist agents, since they would not have enough information for making decisions about norm compliance and a great part of norms would be ignored.

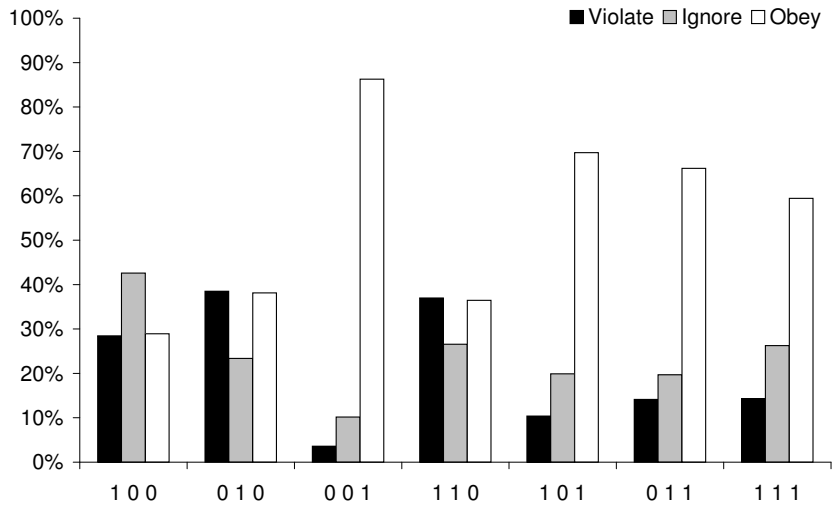

Figure 5.11: Percentage of instances that belong to each willingness category on average when the number of goals is 10

5.4.2.4 Explanatory Relationships

The last simulation consists in varying the number of explanatory relationships that an agent knows. Figures 5.14, 5.15 and 5.16 show the results obtained when the number of explanatory relationships is 10, 20 and 40, respectively. According to the definitions of the willingness func-

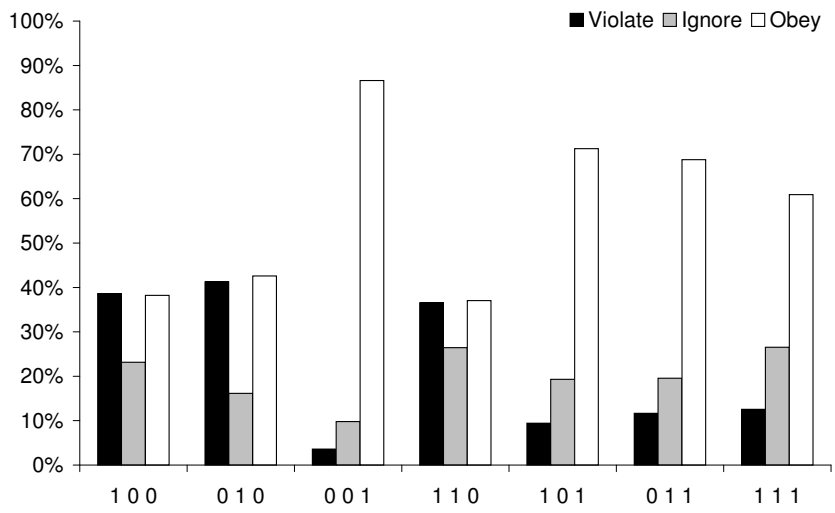

Figure 5.12: Percentage of instances that belong to each willingness category on average when the number of goals is 50

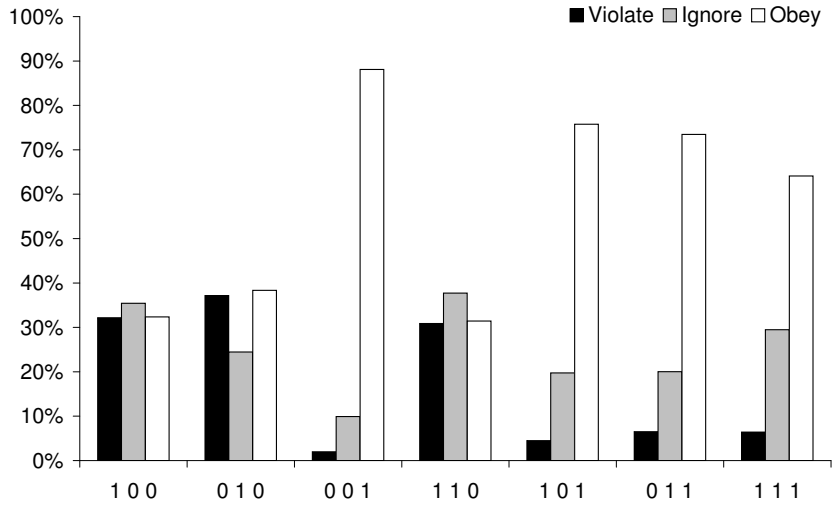

Figure 5.13: Percentage of instances that belong to each willingness category on average when the number of goals is 100

tions, only those agents that consider emotions are affected by the explanatory relationships. When the number of explanatory relationships is low (see Figure 5.14) agents that consider emotions have very few information for violating norms due to their bad consequences so the number of obeyed norms increases. This increase is higher in agents that only consider emotions ($w_{interest} = 0$, $w_{expectation} = 0$ and $w_{emotion} = 1$). As the number of explanatory relationships increases (see Figure 5.15), the number of obeyed norms decreases lightly (e.g. with 10 explanatory relationships the 88.99% of norms are obeyed, whereas with 40 explanatory relationships the 86.89% of norms are obeyed). When the number of explanatory relationships is equal or higher than 40 (see Figure 5.16) the performance of emotional agents is quite similar to the

average results.

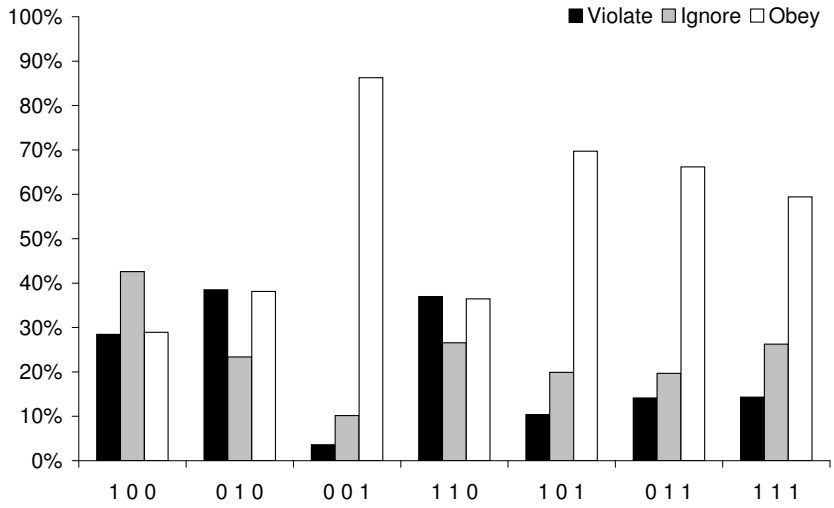

Figure 5.14: Percentage of instances that belong to each willingness category on average when the number of explanatory relationships is 10

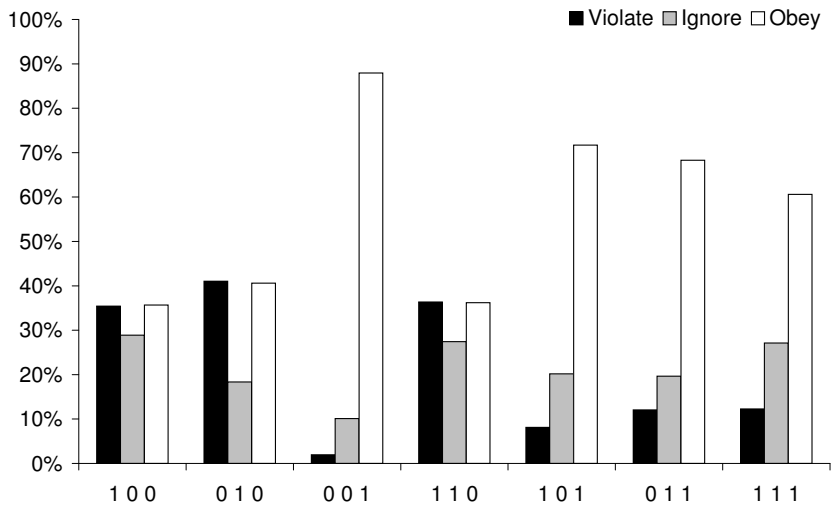

Figure 5.15: Percentage of instances that belong to each willingness category on average when the number of explanatory relationships is 20

5.4.3 Discussion

As shown by the results provided in this section, the deliberation mechanism proposed in this chapter allows agents to make decisions about norm compliance autonomously. However, the behaviour of an agent depends on the willingness factors that it considers and, as shown by the experimental results, it is predictable to some degree. In this way, the designers of MAS

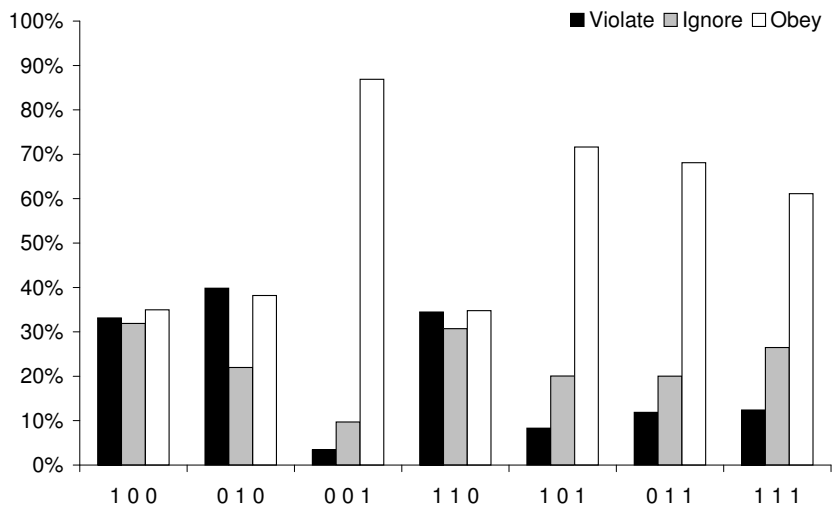

Figure 5.16: Percentage of instances that belong to each willingness category on average when the number of explanatory relationships is 40

or the human user of the *assistant* agent can decide the behaviour of agents according to the functionality that is required.

As illustrated by these results, the emotional factor sustains compliance with norms that do not have a direct effect on the agent's goals. However, to avoid that all norms are blindly followed when norms have a high acceptance degree, emotions should be combined with other factors such as expectations or self-interest.

Finally, it should be noted that improving the agent capabilities for making decisions about norm compliance obviously comes at an additional temporal cost. Specifically, Normative BDI agents must evaluate each instance against its desire set for calculating the self-interest and the expectation factors. To calculate the prospect emotion, agents must evaluate each instance against its desire and belief sets to determine the desirability of the repercussions of instances. Finally, in the calculation of the attribution emotion multiple substitutions are applied to determine the norms that are a generalization of each instance. This step may be computationally expensive if the number of instances, norms and substitutions is high. However, this problem can be easily avoided if instances are annotated with the norm that has created the instance and the attribution emotion is simply calculated as the acceptance degree of this norm[11].

[11]This simplification does not take into account that an instance can be seen as a particularization of more than one norm. Thus, it assumes that norms that generate similar instances have similar acceptance degrees.

5.5 Contributions

This chapter answers two main questions. The first one is related to the possibility of developing norm-autonomous agents that consider emotional criteria in their decisions about norm compliance. In response to this issue, this chapter describes how n-BDI agents consider both their preferences and the norm repercussions when they determine their willingness to comply with norms. The repercussion of norms is not only defined in terms of the utility of norms and the economic cost (vs. benefit) of the sanctions (vs. rewards), but also in terms of the social repercussion of norms (i.e., emotional factors). Specifically, agents are endowed with mechanisms for anticipating the emotions that will be elicited if the norms are transgressed. Moreover, the way in which agents combine rational and emotional factors allow different personalities to be modelled. As far as we are concerned, this is the first proposal of norm-autonomous agents that considers emotions as a motivation for norm compliance. The second question addressed by this chapter is to determine if the emotional criteria are useful for making decisions about norm compliance. Up to now, decisions about norm compliance only consider the effect of norms on the agents' goals. As illustrated by the experimental results, emotions can explain norm compliance even if norms do not affect directly the agent goals. From these experimental results we can conclude that emotions are one important factor that must be deeply considered in the development of norm-autonomous agents. We believe that these emotional criteria are required in applications such as: social simulation scenarios, environments in which humans and agents interact in a realistic way, scenarios in which humans delegate tasks to personal software agents, and so on.

5.6 Conclusions

This chapter is focused on the development of reasoning mechanisms for allowing n-BDI agents to take into account deontic norms. These deontic norms are the extrinsic motivations of agents. Specifically, this chapter describes a deliberation mechanism for allowing n-BDI agents to determine their willingness to comply with norms according to rational and emotional factors. The way in which rational and emotional factors are combined allows different personalities to be modelled. In the next chapter we extend the n-BDI architecture with capabilities for reasoning about constitutive norms. Specifically, the reasoning mechanisms proposed in the next chapter allow agents to keep track of the institutional state given that they are allocated

in the real world.

Chapter 6

Norm-based Expansion: Reasoning About Constitutive Norms

Agents may become members of different institutions along their life and, they might even belong to different institutions simultaneously. For these reasons, agents need capabilities that allow them to determine the repercussion that their actions would have in the different institutions. This anchorage between the real word, in which agents' interactions and actions take place, and the institutional world is defined by means of *constitutive* norms. *Constitutive* norms are used for establishing social institutions which give rise to new types of facts that only make sense within the institution. This chapter considers the role of constitutive norms inside the n-BDI architecture that has been proposed in the previous section.

This chapter is structured as follows: Section 6.1 contains an introduction to this chapter; Section 6.2 illustrates how n-BDI agents reason about constitutive norms. This reasoning process has been applied into a case study in Section 6.3. Section 6.4 describes the experiment that we carried out. Finally, contributions and conclusions are contained in Sections 6.5 and 6.6, respectively.

6.1 Introduction

The term norm has been traditionally used for referring to deontic norms (see Definition 3.2.1) that define patterns of behaviour aimed at regulating the actions of software agents and the interactions among them. However, norms are not only deontic prescriptions, but they also establish social institutions which give rise to new types of facts. These facts are named

institutional facts since they only make sense within institutions [Sea05]. This type of norms is known as *constitutive norms* since they create the institutional reality; i.e., they regulate the creation of institutional facts. *"A piece of paper made by a national bank counts as money"* is a well known example of constitutive norm. One of the most well-known and referred proposals on constitutive norms is made by Searle in [Sea69]. In this work Searle proposes a classification of norms into "regulative" and "constitutive" ones. According to Searle's definition, constitutive norms define the *counts-as* relationship. This relationship defines how the institutional reality (i.e., the institutional facts) is built in terms of actions or state of affairs occurring in the real world (i.e., brute facts).

Traditionally, constitutive norms have been used as bricks for building the ontology of institutions. These contextual ontologies define a link between abstract concepts in which deontic norms are defined to the real facts that take place in the application domain. Thus, constraints aimed at achieving the desired behaviour (i.e., the deontic norms) are specified in at higher abstract level (i.e., in terms of institutional facts) in order to allow different situations to be controlled through a reduced set of constraints [VS03, Ald09]. We claim that constitutive norms are not simple bricks for building institutional ontologies used on the definition of deontic norms. As a consequence, norm aware agents need to consider constitutive norms not only for translating abstract deontic norms into concrete ones, but also for selecting the most suitable actions according to their goals and the institutional repercussions. Several proposals have been made in order to define agents provided with norm reasoning capabilities [BDH$^+$01, KN03, SST06]. In particular, these works are aimed at describing how deontic norms, which define regulations or constraints on agents' behaviours, are considered by agents. However, the role of constitutive norms in agent reasoning has not been taken into account by these previous works. Therefore, there is a lack of elaborated decision making procedures which consider the role of constitutive norms inside agents' minds.

In this thesis we propose to endow agents with an explicit representation of constitutive norms that brings them the possibility of reasoning about the interpretations of their actions in the different institutions. Specifically, the main contribution of this chapter consists in allowing agents to consider the impact of their actions on the institutions and making decisions accordingly.

6.2 Norm-based Expansion for Constitutive Norms: Proposition Generation

The norm expansion is the process of acceptance of a set of norms. Basically, the agent goes through a process of understanding why they are of value or why they make sense, until norms are finally accepted as the agent own viewpoint. As stated in the previous chapter, after deontic norms are instantiated inside the NCC, these deontic instances must be used in order to extend the agent's desires according to norms. In case of constitutive norms they are used to extend the agent beliefs and desires. Figure 6.1 illustrates the norm expansion process for constitutive norms. Thereby, agents are able to determine the effect that their actions would have on the institutional state.

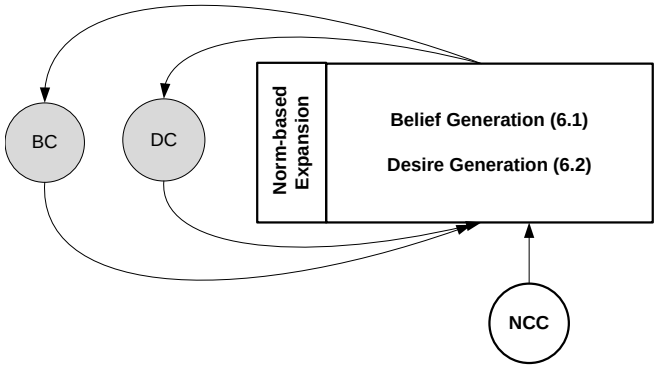

Figure 6.1: Norm-based Expansion for Constitutive Norms in the n-BDI Architecture. Contexts are represented as circles, whereas sets of bridge rules that perform similar tasks are represented as boxes in which there are input links, which are the premises of bridge rules, and output links, which represent the conclusions. Gray circles correspond to the basic architecture that has been defined in previous works [CGS11]. The normative extensions are the white elements.

Next, the concrete bridge rules for creating beliefs and desires from constitutive norms are provided:

- *Belief Generation Bridge Rule.* Informally, a constitutive norm is a rule which determines in which circumstances a brute fact *counts-as* institutional fact. The next bridge rule transforms a belief which is affected by a constitutive norm as follows (see Figure 6.1 Bridge Rule 6.1):

$$\frac{\begin{array}{c} NCC : instance(\langle I', self, A', E', BF', IF' \rangle, \rho_{NCC}), \\ BC : (\mathcal{B}\ BF'', \rho_{BF''}) \end{array}}{BC : (\mathcal{B}\ IF'', f_{expansion}(\rho_{BF''}, \rho_{NCC}))} \tag{6.1}$$

If an agent considers that a constitutive norm has been instantiated ($instance(\langle I', self, A', E', BF', IF' \rangle, \rho_{NCC})$) and the basic fact ($BF'$) affected by the constitutive norm is an agent belief, then a new belief will be inferred corresponding to the new institutional fact ($\mathcal{B}\ IF'', f_{expansion}(\rho_{BF''}, \rho_{NCC})$)[1].

The certainty degree assigned to IF'' represents the certainty of the declaration of the institutional fact taking into account that agents have no perfect observations of the word. Specifically, the certainty degree of the new belief depends on the certainty degree of the brute fact and the degree in which the norm is considered as relevant. Thus, the $f_{expansion}$ function combines the certainty about the activation of the norm (ρ_{NCC}) and the certainty about the occurrence of the brute fact ($\rho_{BF''}$). Both conditions, the activation of the norm and the occurrence of the brute fact, are required for creating a new belief. Therefore, $f_{expansion}$ is defined as a symmetric sum as follows:

$$f_{expansion}(\rho_{BF''}, \rho_{NCC}) = \frac{\rho_{BF''} * \rho_{NCC}}{1 - \rho_{BF''} - \rho_{NCC} + (2 * \rho_{BF''} * \rho_{NCC})}$$

- *Desire Generation Bridge Rule.* Constitutive norms affect desires oppositely to beliefs. Agents' motivations are the basis for determining which actions will be carried out. Since agents have no capabilities for altering the institutional state directly, then constitutive norms define how abstract desires (which are related to institutional facts) can be redefined in terms of brute facts which can be modified by agents.

$$\frac{\begin{array}{c} NCC : instance(\langle I', self, A', E', BF', IF' \rangle, \rho_{NCC}), \\ DC : (\mathcal{D}\ IF'', \rho_{IF''}) \end{array}}{DC : (\mathcal{D}\ BF'', f_{expansion}(\rho_{IF''}, \rho_{NCC}))} \tag{6.2}$$

In this case, if the institutional fact IF', affected by the constitutive norm, is desired by the agent, then a new desire will be inferred corresponding to the concrete fact ($\mathcal{D}\ BF'', f_{expansion}(\rho_{IF''}, \rho_{NCC})$. The $f_{expansion}$ function has been defined a symmetric

[1] BF' may contain free variables. When there is a substitution σ such as the $BF'' = \sigma(BF')$ and $IF'' = \sigma(IF')$ the bridge rule is applied.

sum as follows:

$$f_{expansion}(\rho_{IF''}, \rho_{NCC}) = \frac{\rho_{IF''} * \rho_{NCC}}{1 - \rho_{IF''} - \rho_{NCC} + (2 * \rho_{IF''} * \rho_{NCC})}$$

The main difference between the implementation of constitutive and deontic norms is that deontic norms are motivational (i.e., they create new desires in order to comply with the norms) whereas constitutive norms are a special kind of inference rules for extending the belief and desire theories. Therefore, it has been considered that a constitutive norm does not affect directly the agents' behaviour. So agents have no motivations for considering or ignoring constitutive norms. For this reason, how agents make decisions about accepting constitutive norms does not make sense.

6.3 Case Study

This example shows how agents employ constitutive norms for extending their knowledge base and how constitutive norms affect the decision making process.

6.3.1 Initial Situation

Let us suppose that there are two Spanish agents a and b which are "a couple". In this example, we will focus our attention in agent a. Agent a considers that a couple are two agents that are in love and that live together. However, agents a and b do not live together. Thus, a has a belief corresponding to being a couple with b with a certainty degree equal of 0.5 (i.e., $(\mathcal{B}\ couple(a, b), 0.5)$). Regarding motivations of agent a, let us suppose that it wants to be married with agent b with the highest intensity (i.e. $(\mathcal{D}\ married(a, b), 1)$).

6.3.2 Normative Reasoning Process

Norm Acquisition. Marriage is an institutional fact and agent a does not know the procedure by which it can marry agent b. Thus, it asks to two different lawyers l_1 and l_2 which inform about how it can be done. Thus, agent a executes Norm Opinion bridge rules (see Section 4.3.2) and updates its NAC. Specifically, both l_1 and l_2 sent messages that inform about the existence of a constitutive norm as follows:

$$\langle spain, couple(X,Y), \neg couple(X,Y),$$
$$formalized(marriage, X, Y), married(X,Y)\rangle$$

This norm claims that in Spain if any pair of agents X, Y, which are a couple $couple(X,Y)$, formalize a marriage contract ($formalized(marriage, X, Y)$), then it *counts-as* as they are married ($married(X,Y)$).

Norm Acceptance. Both lawyers inform about the same norm n but they have provided different salience values. Agent a executes the Salience Aggregation bridge rule (Equation 4.2) to determine which is the salience of norm n. Specifically, l_1 is completely sure that n is applicable (i.e., ρ_{l_1}=1), whereas l_2 is not sure (i.e., ρ_{l_2}=0.2). Therefore, the set of opinions is $O = \{1, 0.2\}$ and the similarities between each one of the elements in O are $Sim(O) = \{0.2, 0.2\}$. Reputations of l_1 and l_2 are 0.7 and 0.1, respectively. Thus, the set of reputations is $R = \{0.7, 0.1\}$ and the similarities between reputations are $Sim(R) = \{0.4, 0.4\}$. The conflict raised by each expert is $Conflict = \{0.32, 0.32\}$. Finally the reliability of experts is $Reliability = \{0.48, 0.07\}$. Therefore the salience of this norm is 0.9[2] and new norm predicate is created inside the NAC as follows:

$$norm(\langle spain, couple(X,Y), \neg couple(X,Y),$$
$$formalized(marriage, X, Y, C), married(X,Y,C)\rangle, 0.9)$$

Norm Instantiation. Next, bridge rules for instantiating constitutive norms belonging to the NAC into terms belonging to the NCC are applied (Equation 4.3).

$$\frac{\begin{array}{c} NAC : norm(\langle spain, couple(X,Y), \neg couple(X,Y), \\ formalized(marriage, X, Y), married(X,Y)\rangle, 0.9), \\ BC : (\mathcal{B}\ couple(a,b), 0.5), BC : (\mathcal{B}\ member(self, spain), 1) \end{array}}{\begin{array}{c} NCC : instance(\langle spain, self, couple(a,b), \neg couple(a,b), \\ formalized(marriage, a, b), married(a,b)\rangle, f_{relevance}(0.5, 1)) \end{array}}$$

In this case, variables X and Y of the abstract norm are instantiated by the values a and b, respectively:

$$\langle spain, self, couple(a,b), \neg couple(a,b), formalized(marriage, a, b),$$
$$married(a,b)\rangle$$

[2]
$$\theta_{acquisition} = \frac{0.48 * 1 + 0.07 * 0.2}{0.48 + 0.07} = 0.9$$

is inserted in the NCC. Considering the definition of $f_{relevance}$ as a symmetric sum, then $f_{relevance(0.5,1)} = 1$.

Norm-based Expansion. In this case, according to the set of beliefs and desires, the bridge rule for generating desires from instances is executed as follows (Equation 6.2):

$$\frac{\begin{array}{c} NCC : instance(\langle spain, self, couple(a,b), \neg couple(a,b), \\ formalized(marriage,a,b), married(a,b)\rangle, 1), \\ DC : (\mathcal{D}\ married(a,b), 1) \end{array}}{DC : (\mathcal{D}\ formalized(marriage,a,b), f_{expansion}(1,1))}$$

Considering the definition of $f_{expansion}$ as a symmetric sum, then $f_{expansion}(1,1) = 1$. Thus, a new positive desire is generated inside the NCC:

$$DC : (\mathcal{D}\ formalized(marriage,a,b), 1)$$

Decision Making. After normative bridge rules have been applied for extending the mental theories (i.e., the set of beliefs and desires), bridge rules for making a decision about the next action to perform are considered. Mainly, this process consists in generating plans for reaching the desired state given that the agent knows the existence of actions that could achieve it. For example, agent a knows that a contract C among two agents X and Y is formalized when both agents sign this contract and the contract is registered. Thus, the agent generates different intentions according to all feasible plans and selects one of them to be executed. As a result, agent a formalizes a marriage contract and updates its beliefs accordingly. A belief such as this is inserted into the BC:

$$((\mathcal{B}\ formalized(marriage,a,b), 1))$$

Norm-based Expansion (2nd Iteration). Since the belief base has changed, the bridge rule for the generation of beliefs from instances belonging to the NCC is triggered. In this case, the bridge rule for extending the belief theory (Equation 6.1) will be applied:

$$\frac{\begin{array}{c} NCC : instance(\langle spain, self, couple(a,b), \neg couple(a,b), \\ formalized(marriage, a, b), married(a,b)\rangle, 1), \\ BC : (\mathcal{B}\ formalized(marriage, a, b), 1) \end{array}}{BC : (\mathcal{B}\ married(a,b), f_{expansion}(1,1))}$$

As previously mentioned $f_{expansion}(1,1) = 1$, so then a belief such as $(\mathcal{B}\ married(a,b), 1)$ will be inserted into the NCC. Thanks to this belief, the abstract desire of being married can be retracted, since it has been achieved.

Constitutive Norm Expiration. Let us suppose that agent a and b are no longer a couple $(B\ \neg couple(a,b), 1)$. Thus, the constitutive norm expires and it cannot be applied. However, the belief about the marriage entered into a and b will not be affected. This is logical since institutional facts (like marriage) are not directly controllable by agents.

The marriage example is a metaphor for the definition of agent federations inside institutions. As illustrated by this section, agents are capable of performing those actions that entail the modification of the institutional state (i.e. the creation of federations).

6.4 Experimental Results

This section illustrates experimentally the performance of n-BDI agents with respect to their capabilities for reasoning about constitutive norms. Specifically, we have performed an experiment aimed at determining to what extent our proposal allows agents situated in uncertain environments to keep track of the institutional state. Specifically, we want to determine: whether or not n-BDI agents detect the dynamics of constitutive norms; and whether or not the use of graded logics to represent both mental and normative propositions allows agents to be aware of the institutional state with more precision[3]. To this aim, we have compared the results obtained by n-BDI agents with respect to BDI agents that use classical logics that restrict the number of truth values to only two.

In this experiment, there is a set of agents that is informed by experts about the salience of constitutive norms. Table 6.1 sums up the parameters of the experiment. In this scenario,

[3]For simplicity, we will only focus on detecting the institutional changes; i.e., this experiment only takes into account how agents extend their belief base. As a consequence, the results described in this section only take into account the generation of beliefs from constitutive norms (see Bridge Rule 6.1 in Section 6.2). However, similar results were obtained if we also considered the generation of desires.

Parameter	Value
# of norms	100
# of agents	100
# of simulations	1000
# of iterations	100
# of experts	10
Expert accuracy	$[0, 1]$
Agent accuracy	$[0, 1]$
Agent precision	$[0, 1]$

Table 6.1: Parameters used in the norm expansion experiment

we employed 100 agents. These agents belong to the same institution in which there are 100 different norms. Agents are informed by a set of experts about these norms. The accuracy of each one of the experts to determine the salience of norms ranges randomly within the $[0, 1]$ interval. The higher the accuracy of an expert, the more precise the opinions that the expert provides. Hence, the opinions provided by experts are affected by a random normally-distributed noise. We consider a normally-distributed noise with mean 0.0 and a varying standard deviation depending on the expert accuracy[4]. Finally, n-BDI agents should determine which is the reputation of each expert with respect to their recommendations about norms. Each n-BDI agent has an accuracy degree that ranges within the $[0, 1]$ interval and determines the exactness of the reputations that it calculates. Reputations are also affected by a random normally-distributed noise. Once agents have calculated the salience of constitutive norms, they observe their environment to determine which constitutive norms are relevant to the current situation. When a change in their environment occurs, then agents determine if this change corresponds to a brute fact that is contained in a relevant constitutive norm and the institutional state has changed. To detect changes in the agents' environment, agents are able to observe their environment. However, the exactitude of these observations depends on the agent precision, which is represented as a real within the $[0, 1]$ interval. The highest the precision, the more exact the observations are.

In each simulation, agents are created with random accuracy and precision degrees. Moreover, 10 experts, which have a random accuracy, are created[5]. Agents ask to all experts about the salience of constitutive norms. According to the opinions provided by experts and the

[4]Specifically, we consider the distribution $\mathcal{N} \sim (0, \frac{1-accuracy}{2})$.

[5]Since we want that this experiment is not affected by the capabilities of agents to determine the salience of constitutive norms we have fixed the number of experts to 10

reputations that each agent assigns to experts, the salience is calculated by agents using the R-LOP technique (see Section 4.3.2 in Chapter 4). From that moment on, agents observe their environment and update their belief base accordingly in each iteration. Then, they calculate the relevance of constitutive norms. Finally, they update their beliefs according to the changes that have occurred in the institutional state. Therefore, each agent acts as a binomial classifier that determines which of the institutional facts hold and which ones not. In each iteration we compare the estimation made by agents against the institutional state. Specifically, in each iteration we update: the number of true positives (TP), which is the number of times that an agent considers that an institutional fact is true and it is actually true; the number of true negatives (TN), which is the number of times that an agent considers that an institutional fact is not true and it is actually false; the number of false positives (FP), which is the number of times that an agent considers that an institutional fact is true and it is not true; and the number of false negatives (FN), which is the number of times that an agent considers that an institutional fact is false and it is actually true. Each simulation has been repeated 1000 times to support findings.

6.4.1 Agent Implementation

6.4.1.1 n-BDI Agents

As explained in 4.4.2 n-BDI agents apply bridge rules 4.5 and 4.6 to determine which constitutive norms are relevant to the current situation. For example, constitutive norm activation bridge (bridge rule 4.5) has been defined in Section 4.4.2 as follows:

$$\frac{NAC : norm(\langle I, A, E, BF, IF\rangle, \rho_{NAC}), BC : (\mathcal{B}\, A', \rho_{A'}), BC : (\mathcal{B}\, member(AgentID, I'), \rho_{I'})}{NCC : instance(\langle I', AgentID, A', E', BF', IF'\rangle, f_{relevance}(\rho_{A'}, \rho_{I'}))}$$

Then, n-BDI apply bridge rules 6.1 and 6.2 to extend their mental state according to constitutive norms. Specifically, when an agent considers that a constitutive norm is relevant and that the brute fact affected by the norm is true, then it creates a new belief representing the institutional change. The degree of the new belief depends on the relevance of the norm and on the certainty about the brute fact. If the degree of the new belief is very low with respect to the other beliefs of the agent, then the agent ignores it. In this experiment, we

assume that the beliefs about institutional facts are ignored when their certainty is lower than an internalization threshold ($\delta_{internalization}$).

6.4.1.2 BDI Agents

BDI agents use classical logic for the internal representation of cognitive elements and constitutive norms. Since BDI agents are situated in the real world, they need to convert the uncertain observations into discrete observations that can be represented as two valued propositions. In this simulation we assume a simple approach in which observations that are perceived with a certainty higher than a threshold ($\delta_{observation}$) are considered as true by BDI agents. The salience of a given norm may be considered as the certainty in which this norm is observed. Thus, only those norms whose salience is higher than $\delta_{observation}$ are taken into account by BDI agents.

For example, the constitutive norm activation bridge rule for BDI agents that use classical logic is expressed as follows:

$$\frac{NAC : norm(\langle I, A, E, BF, IF\rangle, \rho_{NAC}), BC : (\mathcal{B}\, A'), BC : (\mathcal{B}\, member(AgentID, I'), \quad \rho_{NAC} > \delta_{observation}}{NCC : instance(\langle I', AgentID, A', E', BF', IF'\rangle, 1)}$$

The constitutive instance expiration and the belief generation bridge rules for BDI agents are defined in the same way.

6.4.2 Metrics

6.4.2.1 Sensitivity and Specificity.

Sensitivity [BBC+00] relates to the test's ability to identify positive results. Specificity [BBC+00] relates to the ability of the test to identify negative results. They are calculated as follows:

$$Sensitivity = \frac{TP}{TP + FN} * 100 \qquad Specificity = \frac{TN}{TN + FP} * 100$$

These two metrics are constructed using only two numbers out of the four (TP, TN, FP, FN). As a consequence, they are bound to be highly biased in some trivial way. For example two classifiers that obtain the same number of TP and FN will obtain the same sensitivity regardless

of the number of FP obtained by each classifier.

6.4.2.2 Matthews Correlation Coefficient.

The Matthews Correlation Coefficient (MCC) [BBC+00] is used as a measure of the quality of binary classifications. It takes into account true and false positives and negatives and is generally regarded as a balanced measure which can be used even if the classes are of very different sizes. The MCC can be calculated using the formula:

$$MCC = \frac{TP \times TN - FP \times FN}{\sqrt{(TP + FP)(TP + FN)(TN + FP)(TN + FN)}}$$

The MCC is in essence a correlation coefficient between the observed and predicted binary classifications; it returns a value between -1 and +1. A coefficient of +1 represents a perfect prediction, 0 a random prediction and -1 an inverse prediction. While there is no perfect way of describing the confusion matrix of true and false positives and negatives by a single number, the Matthews correlation coefficient is generally regarded as being one of the best such measures.

6.4.2.3 Threshold Estimation.

To determine the most suitable values for the internalization threshold ($\delta_{internalization}$) and the observation threshold ($\delta_{observation}$) we have performed two experiments varying the value of these two thresholds.For example, in the experiment for determining the value of $\delta_{internalization}$ a set of 100 n-BDI agents are created in each simulation. Similarly, the experiment for determining the value of $\delta_{observation}$ a set of 100 BDI agents are created in each simulation. In each simulation of any of the two experiments, agents are informed about the salience of 100 constitutive norms by 10 experts. The reputation of experts and the accuracy of agents range randomly within the $[0, 1]$ interval. In each iteration, agents perceive their environment and estimate which institutional facts hold and which ones not. The estimation made by agents is compared against the institutional state and the number of TP, TN, FP and FN is updated accordingly. Agents are able to perceive their environment along 10 iterations. For each value of the thresholds we have performed 100 simulations. Figure 6.2 shows the MCC with respect to the value of $\delta_{internalization}$. As illustrated by Figure 6.2, the best results are obtained when $\delta_{internalization}$ is 0.2. Therefore, we fixed the internalization threshold to 0.2 in the rest of the experiments. Similarly, Figure 6.3 shows the MCC with respect to the value of $\delta_{observation}$. As illustrated

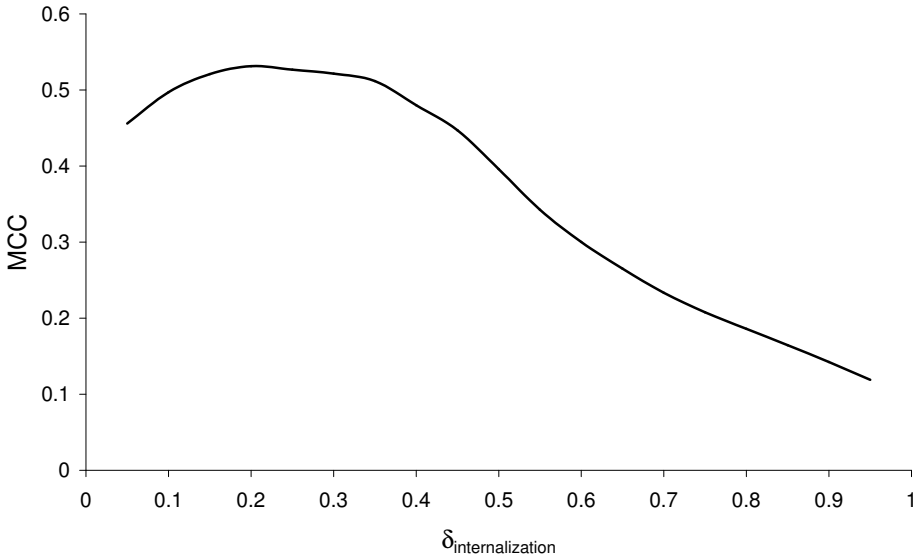

Figure 6.2: MCC with respect to the internalization threshold ($\delta_{internalization}$)

by Figure 6.3, the best results are obtained when $\delta_{observation}$ is 0.25. Therefore, we fixed the observation threshold to 0.25 in the rest of the experiments.

6.4.3 Results

As previously mentioned, the capabilities of n-BDI and BDI agents to infer the institutional state are evaluated by considering the number of TP, TN, FP and FN that they made when they estimate which institutional facts hold and which ones not. Each simulation has been repeated 1000 times to support findings. Table 6.2 shows the Sensitivity, the Specificity and the MCC achieved by n-BDI agents and BDI agents. In light of these results, we can conclude that the n-BDI architecture allows agents to keep track of the institutional state with more precision. Specifically, n-BDI agents have better capabilities for detecting when an institutional fact does not hold (i.e., a high sensitivity means that if an agent determines that a an institutional fact does not hold, then there is a a high probability that the institutional fact does not hold in the institution) than BDI agents. Besides that, n-BDI agents obtain a higher specificity, which means that n-BDI agents identify which institutional facts hold more precisely. On average, the MCC obtained by BDI agents is 0.41. The MCC obtained by n-BDI agents on average is 0.53. The improvement on the MCC achieved by n-BDI agents is 29.27%[6] when they estimate which

[6]The improvement on the MCC is calculated as:

$$\frac{0.53 - 0.41}{0.41} * 100 = 29.27\%$$

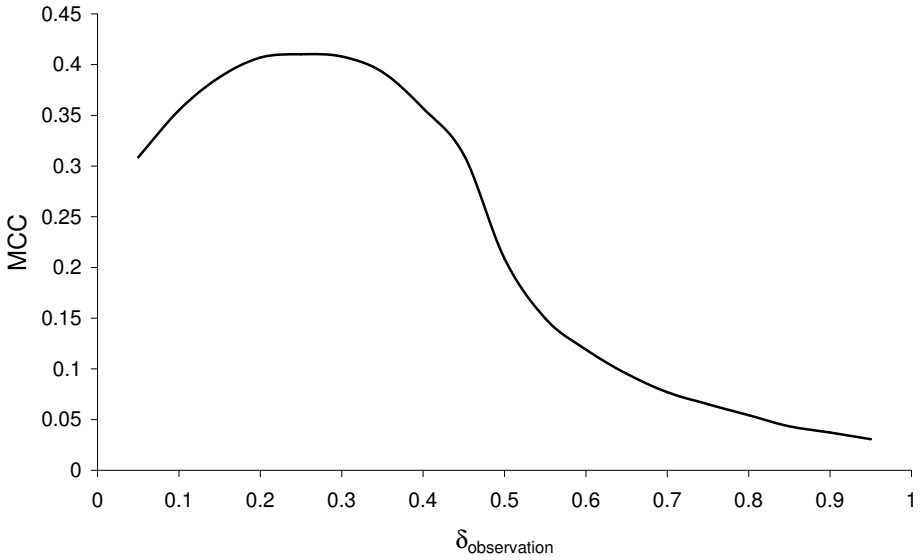

Figure 6.3: MCC with respect to the observation threshold ($\delta_{observation}$)

Agent Type	*Sensitivity*	*Specificity*	*MCC*
n-BDI	$72 \pm 0.12\%$	$93.98 \pm 0.56\%$	0.53 ± 0.03
BDI	$65.27 \pm 2.86\%$	$91.02 \pm 0.27\%$	0.41 ± 0.02

Table 6.2: 95% confidence interval for the Sensitivity, the Specificity and the MCC achieved for each type of agent.

institutional facts hold and which ones not. Hence, we can conclude that the use of graded logics for representing both mental and normative propositions allows agents to keep track of the institutional state with more precision.

6.5 Contributions

In the Artificial Intelligence field, the modelling of the *counts-as* relationship is introduced by Jones and Sergot in [JS96]. From that moment on, several variations of the *counts-as* operator have been proposed. For example, in [GD05] Grossi and Dignum propose an alternative definition of the *counts-as* connective for dealing with non-monotonicity. In [GMD06], Grossi et al. provide semantic interpretation of the *counts-as* relationship by means of modal logic. In particular, the *counts-as* can be interpreted as statements that create general *classifications* that hold in any situation or they can be interpreted as rules aimed at *constituting* or defining contexts in which counts-as hold. According to this meaning, in [GAVSD06] constitutive norms

are used as bricks for building the ontology of institutions. These contextual ontologies define a link between abstract concepts to the real facts that take place in the application domain.

In [Ald09, VS03] constitutive norms are an abstraction mechanism that allows the definition of abstract regulative norms used in the specification and implementation of norms inside electronic institutions. These two works deal with the implementation of constitutive norms from an institutional perspective. They propose that the institution should translate abstract regulative norms into concrete ones making use of the ontology defined by constitutive norms. These concrete regulative norms are expressed in terms of work domain facts which are controllable by the institution infrastructure. Similarly, in [AÁNDVS10] an implementation of constitutive norms to relate abstract organizational specifications and norms to concrete situations that take place in the real world is proposed.

A noteworthy work on constitutive norms is the proposal of Boella et al. in [BvdT04a]. In this work, they define a formal model of *Normative MAS* (NMAS) in which the coordination and cooperation is achieved by means of constitutive and deontic (regulative according to Boella et al. terminology) norms. In addition, they use the metaphor of NMAS as agents, thus the NMAS have mental attitudes. In this sense, constitutive norms are not modelled as operative constraints of an institution but as beliefs of the normative agent, whereas deontic norms are the goals of the normative agent. In this proposal, Boella et al. use constitutive norms for describing the legal consequences of actions in the normative system [BvdT05b]. Thus, *metanorms* that define legal procedures for the definition of the normative system (i.e. the norm change procedures) are also constitutive. The work described in [BBT08] details how reasoning about constitutive norms can be done from an institutional perspective. In particular, this work proposes a mechanism for analysing and characterizing the notions of redundancy and equivalence of normative systems formed by both constitutive and deontic norms.

As far as we are aware, the problem of how norm aware agents take constitutive norms into consideration has not been considered by the existing literature. The work of Grossi et al. [GAVSD06] mentions that there is a need for mechanisms for allowing agents to consider constitutive norms. Similarly, in [AÁNDVS10] it is pointed out that constitutive norms may be used by agents to determine normative consequences of actions and determine their future actions according to norms. This usage of constitutive norms as an instrument for allowing deontic norms to be defined in an abstract way, making use of institutional facts that may be translated into different brute facts according to each concrete situation, is also supported

by our proposal. It will be illustrated by means of the following example: in general any highway code contains a norm that forbids agents to commit a driving offence. According to our proposal the prohibition will be translated into a negative desire ($\mathbf{D}\neg drivingOffence$). In most countries, to drive exceeding 50km/h inside the city boundaries count-as a driving offence:

$$\langle spain, inTown(T), \neg inTown(T), \\ exceeed(50), drivingOffence\rangle \qquad \text{(Driving Offence Norm)}$$

When the agent enters a city the constitutive norm becomes active and according to bridge rules for internalising constitutive norms a new negative desire will be inferred ($\mathbf{D}\neg driveFasterThan(50)$). This negative desire will allow the *assistant* agent to avoid those plans in which the speed exceeds 50km/h and that will violate the highway code.

Our thesis is that constitutive norms are not simple bricks for building institutional ontologies used on the definition of deontic norms. As a consequence, norm aware agents need not only to consider constitutive norms for translating abstract deontic norms into concrete ones, but also they must have an explicit representation of constitutive norms. Thus, they would be able to reason about the impact that their behaviour should have on the institutional state.

6.6 Conclusions

In this chapter, we face the norm reasoning problem from the agent point of view emphasizing the role of constitutive norms on agent reasoning processes. The main contribution of the work described in this chapter is to allow n-BDI agents to reason about constitutive norms. Moreover, we have evaluated the capabilities of n-BDI agents to keep track of the institutional state given that they are allocated in the real world. The conclusion of these experiments is that the use of graded logics allow n-BDI agents to reason about constitutive norms with more precision.

However, the set of constitutive norms considered by an agent might be in conflict, since these norms belong to different institutions or normative spaces. Thus, the consideration of coherence for resolving conflicts and inconsistencies among norms and mental propositions is an interesting issue that will be addressed in the next chapter.

Chapter 7

Coherence-based Contraction

The previous chapters have described how the cognitive elements of agents are extended with propositions derived from deontic and constitutive norms. These new propositions might be in conflict with existing ones. Hence, agents should resolve contradictions before making a decision about which action to perform. The coherence-based contraction process, which is described in this chapter, solves the existence of conflicting propositions by calculating and selecting those propositions that maximize the coherence of the cognitive elements present in the agent theory. This chapter is organized as follows: Section 7.1 contains a brief introduction to this chapter; Section 7.2 describes the main principles of coherence theory; Section 7.3 details how this theory has been used in multi-context graded BDI agents; Section 7.4 details the use of coherence in n-BDI agents; Section 7.5 shows an example of the use of coherence in n-BDI agents; Section 7.6 summarises the main contributions of this chapter; and Section 7.7 concludes this chapter.

7.1 Introduction

The *assistant* agent proposed in this thesis builds or searches for feasible routes (that achieve some of the positive desires) that satisfy preconditions (according to its uncertain knowledge of the world) and avoid undesired postconditions (negative desires). As mentioned in Section 4.2.1, the beliefs of the *assistant* agent are propositions that represent the world in which it is situated as well as explanation relationships between beliefs. Thus, the *assistant* agent has primitive beliefs and other ones that can be inferred. Moreover, the *assistant* agent may have beliefs that have been derived from constitutive norms. Since the *assistant* agent is situated in an uncertain environment, it is possible that it has contradictory or conflicting beliefs. For

example, the *assistant* agent may have different beliefs about the road condition that can sustain that the road is unsuitable for driving (e.g., there is heavy rain) and that the road surface is dry. Similarly, the desire context of the *assistant* agent contains propositions that represent the user preferences or goals as well as facilitation relationships between goals. Moreover, the *assistant* agent also has external desires that have been created out of deontic and constitutive instances. As in case of beliefs the set of propositions that are contained in the DC can be inconsistent. The activation and the expiration of norms are sustained by the set of beliefs. In the uncertain situation mentioned above, the *assistant* agent must consider those general norms that are applied when there is heavy rain as those ones that are active only if the road is dry. Norms establish a link among beliefs, instances and desires. Thus, before the *assistant* agent searches for plans or builds new ones (i.e., it generates new traffic routes) it is necessary to resolve conflicts among the belief, desire and instance sets. Therefore, the *assistant* agent needs to determine what norms must be considered in this situation, determining a set of coherent desires and searching for routes according to these desires. The resolution of mental and normative conflicts based on a coherence-maximization approach is explained below.

7.2 Coherence Theory

In [Tha00] Thagard claims that coherence is a cognitive theory whose main purpose is the study of associations; i.e., how pieces of information influence each other by imposing a positive or negative constraint over the rest of information. Thagard proposes the implementation of the abstract theory of coherence as a maximization constraint satisfaction problem. Thus according to Thagard's formalization, a coherence problem is modelled by a graph: nodes represent pieces of information; edges are the positive or negative constraints among information; and each edge has a weight expressing the strength of the coherence or incoherence relationship. The formal definition of a coherence graph is provided below.

Definition 7.2.1 (Coherence Graph [Tha00]) *A coherence graph is an edge-weighted undirected graph* $g = \langle V, E, \zeta \rangle$ *where:*

- V *is a finite set of nodes representing pieces of information;*
- $E \subseteq V^2$ *is a finite set of edges representing the coherence or incoherence between pieces of information;*

- $\zeta : E \rightarrow [-1, 1]$ *is the coherence function that assigns a value to the coherence between pieces of information.*

Maximizing the coherence is the problem of partitioning nodes into two sets (accepted A and rejected $V \setminus A$) which maximizes the strength of the partition, which is the sum of the weights of the satisfied constraints. Next, the formal definitions of satisfied constraints and the strength of a partition are provided.

Definition 7.2.2 (Satisfied Constraints [Tha00]) *Given a coherence graph $g = \langle V, E, \zeta \rangle$ and a partition $(A, V \setminus A)$ of V, the set of satisfied constraints $C_A \subseteq E$ is given by:*

$$\begin{aligned} C_A = \quad & \{(v, w) \in E \mid v \in A \textit{ iff } w \in A, \textit{ when } \zeta(v, w) \geq 0\} \cup \\ & \{(v, w) \in E \mid v \in A \textit{ iff } w \notin A, \textit{ when } \zeta(v, w) < 0\} \end{aligned}$$

All other constraints are said to be unsatisfied.

Definition 7.2.3 (Strength of a Partition [Tha00]) *Given a coherence graph $g = \langle V, E, \zeta \rangle$ the strength of a partition $(A, V \setminus A)$ of V is given by:*

$$\sigma(g, A) = \sum_{(v,w) \in C_A} \frac{|\zeta(v, w)|}{|E|}$$

Definition 7.2.4 (Coherence [Tha00]) *Given a coherence graph $g = \langle V, E, \zeta \rangle$ the coherence of g is given by:*

$$\kappa(g) = \max_{A \subseteq V} \sigma(g, A)$$

If for some partition $(A, V \setminus A)$ of V, the strength of the partition is maximal then the set A is called the accepted set and $V \setminus A$ the rejected set of the partition.

Coherence can be understood in terms of maximal satisfaction of multiple constraints. Thus, the coherence problem consists of dividing a set of elements into accepted and rejected sets in a way that satisfies the most constraints. These elements may be concepts, propositions, parts of images, goals, actions, and so on. According to the nature of these elements different types of coherence can be defined. For example, semantic coherence analyses the relationships among propositions according to their meaning.

In the n-BDI proposal, relationships among mental and normative propositions are defined in terms of inference and bridge rules. Thus, we will focus on deductive coherence, which studies the coherence among logical propositions that belong to a deductive system. Next, deductive coherence principles are explained in detail.

7.2.1 Deductive Coherence

According to Thagard's definition of deductive coherence, a *deductive coherence graph* [Tha00] is a coherence graph whose nodes are propositions and whose pairs of nodes are related by a *deductive coherence function* ζ yielded by propositional logical deduction. There are five principles that establish relations of deductive coherence and that allow the global coherence of a deductive system to be assessed. Given P, Q and $P_1, ..., P_n$ propositions of a deductive system S, the principles of deductive coherence are [Tha00]:

1: Symmetry. Deductive coherence is a symmetric relation.

2: Deduction. If $P_1, ..., P_n$ deduce Q, then:

- **(a)** Any proposition coheres with propositions that are deductible from it. Thus, for each P_i in $\{P_1, ..., P_n\}$, P_i and Q cohere.
- **(b)** Propositions that together are used to deduce some other proposition cohere with each other. For each P_i and P_j in $\{P_1, ..., P_n\}$, P_i and P_j cohere.
- **(c)** The more hypothesis it takes to deduce something, the less the degree of coherence. Thus, in (a) and (b) the degree of coherence is inversely proportional to n.

3: Intuitive Priority. Propositions that are intuitively obvious have a degree of acceptability on their own. Propositions that are obviously false have a degree of rejectability on their own.

4: Contradiction. Contradictory propositions are incoherent with each other.

5: Acceptability. The acceptability of a proposition in a system of propositions depends on its coherence with them.

For this framework to be fully computational, it is necessary to define how a coherence graph can be constructed. Next, how this framework has been applied for calculating coherence in multi-context graded BDI agents is explained.

7.3 Coherence for Multi-context Graded BDI Agents

Once the general notion of deductive coherence has been provided in the previous section, it is necessary to instantiate this general theory into the particular problem of coherence among graded propositions. In [JcSSD10], Joseph proposes a formalisation of the notion of deductive coherence for multi-context graded BDI agents together with mechanisms for calculating the coherence of a set of graded mental attitudes. Next, the formalization of deductive coherence and the mechanisms for calculating coherence are briefly described.

7.3.1 Formalization of Deductive Coherence for Graded Logics

Let $\mathcal{L}$ be a graded logical language and $\vdash$ the inference rules of this language. Thus, $\mathcal{L}$ is formed by expressions such as (α, r); where α is a proposition of a given logic language and $r \in [0, 1]$ is the certainty of this proposition. Finally, let $\bar{0}$ be the falsity constant.

Definition 7.3.1 (Support Function [JcSSD10]) *Let $\mathcal{L}$ be a graded logical language and $\vdash$ the inference rules of this language. Let $\mathcal{T} \subseteq \mathcal{L}$ be a finite theory presentation using graded formulas. A support function $\eta : \mathcal{T} \times \mathcal{T} \to [-1, 1]$ with respect to $\mathcal{T}$ is given by:*

$$\eta(\Phi, \Psi) = \begin{cases} max \begin{cases} \frac{r}{|\Gamma|+1} & \text{where } \Gamma \text{ is the smallest subset of } \mathcal{T} \text{ such that} \\ & \Gamma, \Phi \vdash \Psi \text{ and } \Gamma \nvdash \Psi \text{ and } \Phi \nvdash \text{ and } \Psi = (\alpha, r) \\ \frac{r}{|\Gamma|+2} & \text{where } \Gamma \text{ is the smallest subset of } \mathcal{T} \text{ such that} \\ & \exists (\alpha, r) \in \mathcal{T} \text{ with } \alpha \neq \bar{0} \text{ such that } \Gamma, \Phi, \Psi \vdash (\alpha, r) \\ & \text{and } \Gamma, \Phi \nvdash (\alpha, r) \text{ and } \Gamma, \Psi \nvdash (\alpha, r) \\ \frac{-r}{|\Gamma|+1} & \text{where } \Gamma \text{ is the smallest subset of } \mathcal{T} \text{ such that} \\ & \Gamma, \Phi, \Psi \vdash (\bar{0}, r) \text{ and } \Gamma, \Phi \nvdash (\bar{0}, r) \text{ and } \Gamma, \Psi \nvdash (\bar{0}, r) \end{cases} \\ \text{undefined, otherwise} \end{cases}$$

In order to make coherence a symmetric relationship, the deductive coherence between two propositions is defined by a coherence function as follows:

Definition 7.3.2 (Coherence Function [JcSSD10]) *Let L be a logical language and let $\mathcal{T} \subseteq L$ be a finite theory presentation. Let $\eta : \mathcal{T} \times \mathcal{T} \to [-1, 1] \setminus \{0\}$ be a support func-*

tion with respect to $\mathcal{T}$*. A deductive coherence function* $\zeta : (\mathcal{T})^2 \to [-1, 1] \setminus \{0\}$ *with respect to* $\mathcal{T}$ *is given by:*

$$\zeta(\{\Phi, \Psi\}) = \begin{cases} max\{\eta(\Phi, \Psi), \eta(\Psi, \Phi)\} & \text{if } \eta(\Phi, \Psi) \neq 0 \text{ and } \eta(\Psi, \Phi) \neq 0 \\ \eta(\Phi, \Psi) & \text{if } \eta(\Phi, \Psi) \neq 0 \text{ and } \eta(\Psi, \Phi) = 0 \\ & \text{or undefined} \\ undefined & \text{if } \eta(\Psi, \Phi) = 0 \text{ or undefined and} \\ & \eta(\Phi, \Psi) = 0 \text{ or undefined} \end{cases}$$

The deductive coherence function ζ as defined above satisfies Thagard's principles of deductive coherence. For a demonstration see [JcSSD10].

7.3.2 Building the Coherence Graph

Once the coherence among graded propositions has been formalized, then it is necessary to instantiate this proposal in order to calculate coherence among the cognitive elements of a BDI agent. Specifically, the set of nodes of the coherence graph is formed by those propositions belonging to the mental contexts. Weighed links among propositions belonging to the same context are calculated according to the coherence function ζ that considers the axioms and inference rules of this context. Similarly, bridge rules are employed for setting the coherence degree among propositions belonging to different contexts. Thus, the coherence graph that is formed by propositions that belong to the belief context (BC), desire context (DC) and intention context (IC) is defined as follows:

Definition 7.3.3 (Graph-Join Function [JcSSD10]) *Let* $\{C_i\}_{i=1,...,n}$ *be a family of contexts* ($n > 0$)*, and let* B *be a finite set of bridge rules. The graph-join function* ι_B *is defined as follows: Given a tuple of graphs* $\bar{g} = \langle g_1, ..., g_n \rangle$ *(with* $g_i = \langle V_i, E_i, \zeta_i \rangle$*) then* $\iota_B(\bar{g}) = \langle V, E, \zeta \rangle$ *such that:*

- $V = \bigcup_{1 \leq i \leq n} \{i : \Phi | \Phi \in V_i\}$
- $E = \bigcup_{1 \leq i \leq n} \{\{i : \Phi, i : \Psi\} | \{\Phi, \Psi\} \in E_i\} \cup$

$$\bigcup_{b\in B}\left\{\{i:\Phi, j:\psi\}\;\middle|\;\begin{array}{c} i:\Phi \text{ is a premise of } \pi(b) \text{ and } j:\Psi \text{ is the} \\ \text{conclusion of } \pi(b), \text{ where } \pi \text{ is a most general} \\ \text{substitution, such that,for all premises } k:(A,R) \\ \text{of } b, \pi((A,R)) \in V_k \end{array}\right\} \cup$$
$$\bigcup_{b\in B}\left\{\{i:\Phi, j:\psi\}\;\middle|\;\begin{array}{c} i:\Phi \text{ and } j:\Psi \text{ are premises of } \pi(b), i \neq j, \\ \text{where } \pi \text{ is a most general substitution, such that,} \\ \text{for all premises } k:(A,R) \text{ of } b, \pi((A,R)) \in V_k \end{array}\right\}$$

- $\zeta(\{i:\Phi, i:\Psi\}) = \zeta_i(\{\Phi,\Psi\})$ *and* $\zeta(\{i:\Phi, j:\Psi\})$ *for* $j \neq i$ *is defined with respect to the following support function:*

$$\eta(i:\Phi, j:\Psi) = \begin{cases} max \begin{cases} \frac{r}{|\Gamma|+1} & \text{where } \Gamma \text{ is the smallest subset of } V \text{ such} \\ & \text{that } \exists b \in B \text{ such that } \Gamma \cup \{i:\Phi\} \text{ is the} \\ & \text{set of premises and } j:\Psi \text{ with } \psi = (\alpha, r) \\ & \text{is the conclusion of } \pi(b), \text{ where } \pi \text{ is a} \\ & \text{most general substitution, such that, for} \\ & \text{all premises } k:(A,R) \text{ of } b, \pi((A,R)) \in V_k \\ \frac{r}{|\Gamma|+2} & \text{where } \Gamma \text{ is the smallest subset of } V \text{ such} \\ & \text{that } \exists b \in B \text{ such that } \Gamma \cup \{i:\Phi, j:\Psi\} \\ & \text{is the set of premises and } h = (\alpha, r) \text{ is the} \\ & \text{conclusion of } \pi(b), \text{ where } \pi \text{ is a most general} \\ & \text{substitution, such that, for all premises} \\ & k:(A,R) \text{ of } b, \pi((A,R)) \in V_k \end{cases} \\ \text{undefined, otherwise} \end{cases}$$

More details concerning building the coherence graph can be found in [JcSSD10].

7.4 Coherence for n-BDI Agents

The coherence mechanism described in the previous section allows coherence among mental propositions to be calculated. In this section we propose to extend it for considering the relationships among mental propositions and the mental representation of norms and instances.

The coherence graph in case of n-BDI agents is illustrated in Figure 7.1. Basically this process takes into account the following: i) the beliefs that sustain the activation and expiration of norms and other beliefs that explain or contradict them; ii) the norms that have been instantiated; iii) instances and the conflict relationships among them; and iv) the evaluation of the main goals as well as other goals that potentially facilitate them. Thus, the normative coherence process considers propositions belonging to the BC (i.e., beliefs), the NAC (i.e., the norms[1]), the NCC (i.e., instances) and the DC (i.e., desires).[2]

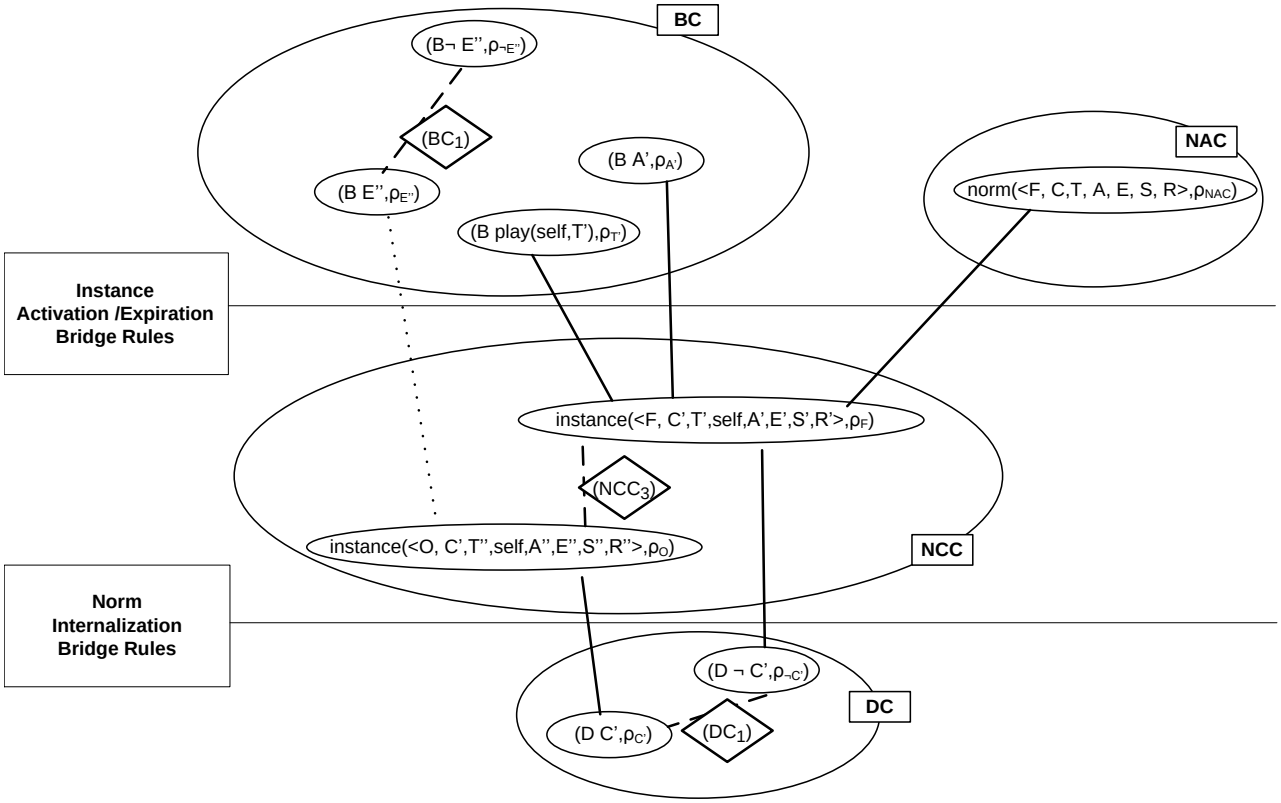

Figure 7.1: Coherence for normative reasoning. This image illustrates the BC, NAC, NCC and DC contexts. The coherence relationships among propositions that belong to a same context are defined by the inference rules of each context. Moreover, incoherence relationships among propositions belonging to a same context (broken lines) are defined by constraints, which are represented as rhombus, that have been added to the BC, NCC and DC. Finally, the coherence and incoherence relationships among propositions belonging to different contexts (represented as bold and dotted lines, respectively) are defined by means of the bridge rules that define the activation and expiration of norms and those bridge rules related to norm internalization.

By considering coherence, we will address three different problems: i) determining norm

[1]Not all norms that have been recognised participate in this process. Only those norms that have been instantiated (i.e., that are relevant) are considered.

[2]Since the reasoning process proposed in this thesis does not affect directly the intentions, the IC context has not been considered for resolving normative conflicts.

activation and deactivation in incoherent states; ii) the resolution of normative conflicts; and iii) deliberating about the most coherent desires and beliefs with respect to norms and their impact on them. In order to use coherence in the n-BDI agent architecture to allow agents to decide which norms will be obeyed, the original proposal, which has been described in the previous section, must be extended with extra constraints. Specifically, the incoherence relationships arise when it is possible to infer the falsity constraint ($\bar{0}$) from a set of propositions. Therefore, it is necessary to define which constraints are used for inferring the falsity constraint in the n-BDI architecture.

7.4.1 Coherence for the BC: Explanatory Constraints

In the case of the BC, the logical deduction has been redefined as an explanation between beliefs (see the definition of the belief context in Section 4.2.1). Thus these explanatory relationships are considered as the basis for calculating the coherence between beliefs. The incoherence relationship among a belief related to a proposition and its negation is defined by means of the addition of an inference rule in the belief context:

$$(BC_1) \quad (\mathcal{B}\gamma, \rho_\gamma), (\mathcal{B}\neg\gamma, \rho_{\neg\gamma}) \vdash (\bar{0}, min(0, 1 - (\rho_\gamma + \rho_{\neg\gamma})))$$

Basically, this scheme means that to belief a proposition (γ) and its negation ($\neg\gamma$) simultaneously is a contradiction ($\bar{0}$) iff the sum of their certainty degrees is higher than 1 [3]. The degree of this contradiction may be informally defined as the "over" certainty assigned to a proposition γ and its negation (i.e., $min(0, 1 - (\rho_\gamma + \rho_{\neg\gamma}))$). Thus, schema BC_1 imposes a restriction over positive and negative beliefs for a same formula. Specifically, BC_1 claims that an agent cannot belief to be in world more than it is not believed. Therefore, it determines that:

$$\rho_\gamma \leq 1 - \rho_{\neg\gamma}$$

For example, the *assistant* agent may believe that the road is being repaired with a certainty of 0.9 —i.e., $(\mathcal{B}\ road(underconstruction), 0.9)$—. This belief is consistent with other perceptions that sustain that the road is not being repaired with a low certainty (i.e., $(\mathcal{B}\ \neg road(under-$

[3] This constraint agrees with the contradiction principle (principle 4) of deductive coherence.

construction), 0.1)). In this case, the degree of $\bar{0}$ is set to 0 (i.e., $min(0, 1-(0.9+0.1)) = 0$) and there is not an incoherence. However, if we consider a situation in which the *assistant* agent believes that the road is not being repaired with a high certainty (i.e., $(\mathcal{B}\neg road(underconstruction), 1)$), then the degree of the inconsistency ($\bar{0}$) will be higher ($min(0, 1-(0.9+1)) = -0.9$) and an incoherence relationship between these two beliefs is defined in the coherence graph.

7.4.2 Coherence for the NCC: Normative Constraints

The notion of coherence is also useful to resolve conflicts among norms. A norm conflict has been defined as a situation in which something is considered as forbidden and obliged[4]. In order to represent incoherence derived from this kind of norm conflict, we add the following inference rule to the NCC:

$$(NCC_1) \quad \begin{array}{l} instance(\langle \mathcal{O}, C', T', AgentID, A', E', S', R' \rangle, \rho_O), \\ instance(\langle \mathcal{F}, C', T'', AgentID, A'', E'', S'', R'' \rangle, \rho_F) \end{array} \quad \vdash (\bar{0}, -min(\rho_O, \rho_F))$$

This consistency constraint represents the conflict among two instances $\langle \mathcal{O}, C', T', AgentID, A', E', S', R' \rangle$ and $\langle \mathcal{F}, C', T'', AgentID, A'', E'', S'', R'' \rangle$ that define opposite deontic relationships (i.e., $\mathcal{O}$ and $\mathcal{F}$) addressed to the same agent ($AgentID$) over the same condition (C'). For a norm conflict to arise, these two instances must be simultaneously active. It implies that both have been activated in some point of the past and they have not expired yet. Therefore, it is true that the time intervals between $A' - E'$ and $A'' - E''$ must overlap. However, it is not necessary to check this explicitly, since two instances are in the NCC only if they are simultaneously active. Since agents may play two or more roles simultaneously, they may be affected by conflicting norms that are addressed to the different roles that they play (i.e. T' and T''). According to the definition of NCC_1, in the case of a conflict between an obligation and a prohibition, the degree of the falsity constant ($\bar{0}$) is assigned a value $-min(\rho_O, \rho_F)$. For example, if an agent is obliged to achieve a given condition C with a certainty 0.5 and it is also forbidden to achieve this condition with a certainty 0.6, then the degree of the incoherence is set to -0.5. However, if the agent is absolutely sure that it is both obliged and forbidden simultaneously, then the norm conflict is stronger and the degree of the incoherence is -1 (i.e., $-min(1, 1) = -1$).

In the n-BDI proposal, permissions are used as a normative operator that define exceptions

[4]Normative constraints are also based on the principle 4 of deductive coherence.

to the application of more general obligation or prohibition norms. Thus, this proposal considers that norms that define something as forbidden and permitted or that oblige to achieve something and that permit not to achieve it are also in conflict:

$$(NCC_2)\quad \begin{array}{c} instance(\langle \mathcal{O}, C', T', AgentID, A', E', S', R' \rangle, \rho_O), \\ instance(\langle \mathcal{P}, \neg C', T'', AgentID, A'', E'', S'', R'' \rangle, \rho_P) \end{array} \quad \vdash (\bar{0}, min(0, 1 - (\rho_O + \rho_P)))$$

$$(NCC_3)\quad \begin{array}{c} instance(\langle \mathcal{F}, C', T', AgentID, A', E', S', R' \rangle, \rho_F), \\ instance(\langle \mathcal{P}, C', T'', AgentID, A'', E'', S'', R'' \rangle, \rho_P) \end{array} \quad \vdash (\bar{0}, min(0, 1 - (\rho_F + \rho_P)))$$

In the case of a conflict between a permission and an obligation or a prohibition, the degree of the falsity constant ($\bar{0}$) is assigned a value $min(0, 1 - (\rho_O + \rho_P))$ or $min(0, 1 - (\rho_F + \rho_P))$, respectively. Thus, if an agent believes that it is forbidden and permitted to achieve a given condition with degrees 0.6 and 0.5, respectively; then the degree of the inconsistency is set to -0.1 (i.e., there is a minor conflict).

7.4.3 Coherence for the DC: Deliberative Constraints

As in the case of the BC, the logical deduction has been used in the DC to represent facilitation and incompatibility constraints between goals. Similarly, the incoherence relationship among conflicting desires is expressed as follows:

$$(DC_1)\quad (\mathcal{D}\gamma, \rho_\gamma), (\mathcal{D}\neg\gamma, \rho_{\neg\gamma}) \vdash (\bar{0}, min(0, 1 - (\rho_\gamma + \rho_{\neg\gamma}))$$

For example, let us consider a situation in which the *assistant* agent wants to drive fast with a desirability 0.75 —i.e., $(\mathcal{D}\ driveFast, 0.75)$—. In this example, this desire represents an internal motivation of the agent that has been derived from the user's desires. However, the agent may have other external motivations that generate a negative desire related to the same proposition. For example, as a consequence of a norm that forbids to drive fast when the road is under construction the next desire may be generated in the $DC : (\mathcal{D}\ \neg driveFast, 0.8)$. This is an inconsistent situation and the degree of $\bar{0}$ is -0.55 (i.e., $min(0, 1-(0.8+0.75)) = -0.55$). As

a consequence, an incoherence relationship between these two desires is defined in the coherence graph.

7.4.4 Coherence Between Contexts: Normative Bridge Rules

Finally, the coherence relationship that exists among propositions belonging to different contexts has been calculated by considering the logical deductions expressed as bridge rules.

Instance Activation and Expiration Bridge Rules. In the case of the n-BDI agent, the instance activation and expiration bridge rules (see Equations 4.3 and 4.4 and Equations 4.5 and 4.6 in Section 4.4.2) allow instances to be connected to beliefs that are related to the activation and expiration conditions. These bridge rules depend on the type of norm that is being considered:

- *Activation and Expiration of Deontic Norms.* Following the mentioned principles of deductive coherence, the coherence relationship between a belief $(\mathcal{B}\ \gamma, \rho_\gamma)$ and a deontic instance $instance(\langle D, C', T', AgentID, A', E', S', R' \rangle, \rho)$ is calculated as follows:

$$\zeta((\mathcal{B}\ \gamma, \rho_\gamma), instance(\langle D, C', T', AgentID, A', E', S', R' \rangle, \rho_{NCC})) = \begin{cases} \rho_{NCC}/2 & \text{if } \gamma = A' \\ \rho_{NCC}/2 & \text{if } \gamma = play(AgentID, T') \\ -\rho_{NCC} & \text{if } \gamma = E' \\ undefined, otherwise \end{cases}$$

  Moreover, the coherence between a deontic norm and an instance that has been created out of this norm is calculated as follows:

$$\zeta(norm(\langle D, C, T, A, E, S, R \rangle, \rho_{NAC}), instance(\langle D, C', T', AgentID, A', E', S', R' \rangle, \rho_{NCC})) = \rho_{NAC}$$

  Notice that the deontic instance activation bridge rule (Section 4.4.2 see Equation 4.3) was defined as:

$$\frac{\begin{array}{c} NAC : norm(\langle D, C, T, A, E, S, R\rangle, \rho_{NAC}), BC : (\mathcal{B}\, A', \rho_{A'}), \\ BC : (\mathcal{B}\, play(AgentID, T'), \rho_{T'}) \end{array}}{NCC : instance(\langle D, C', T', AgentID, A', E', S', R'\rangle, f_{relevance}(\rho_{A'}, \rho_{T'}))}$$

Thus, both believes $(B\, A', \rho_{A'})$ and $(B\, play(AgentID, T'), \rho_T)$ and the norm $(norm(\langle D, C, T, A, E, S, R\rangle, \rho_{NAC}))$ infer the instance. However, the degree of the instance is only determined by the two beliefs; i.e., the degree of the instance is defined as the symmetric sum among $\rho_{A'}$ and $\rho_{T'}$. Thus, both beliefs cohere with the instance in the same manner. Accordingly, the coherence between a deontic instance (i.e., the deduced proposition) and any of the beliefs that sustain its activation (i.e., the hypothesis) are defined as the half of the instance relevance (ρ_{NCC}), since two hypothesis are required for making the deduction[5]. Moreover, the deontic norm $(norm(\langle D, C, T, A, E, S, R\rangle, \rho_{NAC}))$ is necessary to infer the instance but it does not determine the relevance of the instance. Therefore, the coherence among a deontic norm and their instances is defined as the salience of the norm (ρ_{NAC})[6].

The instance expiration bridge rule (Section 4.4.2 see Equation 4.4) was defined as follows:

$$\frac{NCC : instance(\langle D, C', T', AgentID, A', E', S', R'\rangle, \rho_{NCC}), BC : (\mathcal{B}\, E', \rho_E)}{NCC : instance(\langle D', C', T', AgentID, A', E', S', R'\rangle, f_{expiration}(\rho_{NCC}, \rho_E))}$$

Since this bridge rule reduces the certainty of a deontic instance when the agent has a belief about its expiration, then the belief about the expiration of the instance incoheres with the instance. Specifically, the degree of the coherence between a deontic instance and a belief that sustains its expiration is defined as minus the relevance of the instance, since in this case one hypothesis is required for deducting that the norm is not active[7].

- *Activation and Expiration of Constitutive Norms.* Following the mentioned principles of deductive coherence, the coherence relationship between a belief $(\mathcal{B}\ \gamma, \rho_\gamma)$ and a constitutive instance $instance(\langle I', AgentID, A', E', BF', IF'\rangle, \rho_{NCC})$ is calculated as follows:

[5]This agrees with principles 2a and 2c of deductive coherence.
[6]This agrees with principle 2a of deductive coherence
[7]This coheres with principle 4 of deductive coherence

$$\zeta((\mathcal{B}\ \gamma, \rho_\gamma), instance(\langle I', AgentID, A', E', BF', IF'\rangle, \rho_{NCC})) = \begin{cases} \rho_{NCC}/2 & \text{if } \gamma = A' \\ \rho_{NCC}/2 & \text{if } \gamma = member(AgentID, I') \\ -\rho_{NCC} & \text{if } \gamma = E' \\ undefined, otherwise \end{cases}$$

Moreover, the coherence between a constitutive norm and an instance that has been created out of this norm is calculated as follows:

$$\zeta(norm(\langle I, A, E, BF, IF\rangle, \rho_{NAC}), instance(\langle I', AgentID, A', E', BF', IF'\rangle, \rho_{NCC})) = \rho_{NAC}$$

Notice that the constitutive instance activation bridge rule (Section 4.4.2 see Equation 4.5) was defined as:

$$\frac{\begin{array}{c} NAC : norm(\langle I, A, E, BF, IF\rangle, \rho_{NAC}), \\ BC : (\mathcal{B}\, A', \rho_{A'}), BC : (\mathcal{B}\, member(AgentID, I'), \rho_{I'}) \end{array}}{\begin{array}{c} NCC : instance(\langle I', AgentID, A', E', BF', IF'\rangle, \\ f_{relevance}(\rho_{A'}, \rho_{I'})) \end{array}}$$

And the constitutive instance expiration bridge rule (Section 4.4.2 see Equation 4.6) was defined as:

$$\frac{NCC : instance(\langle I', AgentID, A', E', BF', IF'\rangle, \rho_{NCC}), BC : (\mathcal{B}\, E', \rho_{E'})}{NCC : instance(\langle I', AgentID, A', E', BF', IF'\rangle, f_{expiration}(\rho_{NCC}, \rho_{E'}))}$$

Thus, the coherence among a constitutive norm and their instances is defined according to the principles of deductive coherence as in case of deontic norms.

Norm-Based Expansion Bridge Rules. The coherence among instances and those mental propositions that are inferred from them is defined considering the norm internalization bridge

rules.

- *Norm Internalization.*

$$\zeta((\mathcal{D}\ \gamma, \rho_\gamma), instance(\langle D, C', T', self, A', E', S', R'\rangle, \rho_{NCC})) = \begin{cases} \rho_\gamma & \text{if } D = \mathcal{O} \text{ and } \gamma = C' \\ \rho_\gamma & \text{if } D = \mathcal{F} \text{ and } \gamma = \neg C' \\ undefined, otherwise \end{cases}$$

In this case, the bridge rule for internalizing deontic instances (Section 5.2 see Equations 5.1 and 5.2) infers both positive and negative desires from obligation and prohibition instances, respectively. For example, the bridge rule that internalizes obligations was defined as follows:

$$\frac{NCC : instance(\langle \mathcal{O}, C', T', self, A', E', S', R'\rangle, \rho_{NCC}) \wedge \theta_{will} > \delta_{compliance}}{DC : (\mathcal{D}\ C', f_{internalization}(\rho_{NCC}, \theta_{will}))}$$

Thus, the obligation instance infers the normative desire. The coherence between an obligation instance and the desire that is deductible from it is defined as the desirability of the new desire (i.e., only one hypothesis is required for inferring the desire)[8].

- *Proposition Generation.*

$$\zeta((\mathcal{M}\ \gamma, \rho_\gamma), instance(\langle I', self, A', E', BF', IF'\rangle, \rho_{NCC})) = \begin{cases} \rho_\gamma & \text{if } \mathcal{M} = \mathcal{B} \text{ and } \gamma = IF'' \\ \rho_\gamma & \text{if } \mathcal{M} = \mathcal{D} \text{ and } \gamma = BF'' \\ undefined, otherwise \end{cases}$$

In this case, the bridge rules for generating propositions according to constitutive norms (Section 6.2 see Equations 6.1 and 6.2) infers both beliefs and desires. For example, the bridge rule that generates beliefs was defined as follows:

[8]This agrees with principles 2a and 2c of deductive coherence

$$\frac{NCC : instance(\langle I', self, A', E', BF', IF' \rangle, \rho_{NCC}), \quad BC : (\mathcal{B}\ BF'', \rho_{BF''})}{BC : (\mathcal{B}\ IF'', f_{expansion}(\rho_{BF''}, \rho_{NCC}))}$$

Thus, the constitutive instance infers the new belief. The coherence between a constitutive instance and the belief that is deductible from it is defined as the certainty of the new belief (i.e., only one hypothesis is required for inferring the belief)[9].

The coherence among propositions belonging to different contexts has been defined considering the salience of norms, the relevance of instances and the internalization degree that is calculated for each instance.

7.4.5 Coherence Maximization

Using the coherence function defined, a coherence graph can be constructed (see Figure 7.2 for an example). Then, a maximising partition over this graph is calculated following Joseph's proposal. Then, the set of propositions (i.e., the state of mind) is revised in order to consider only those propositions that maximize coherence. As a result, some instances might be deleted and the corresponding normative desires would not be considered for the decision making process. This does not imply the fulfilment of the remaining instances. In fact, it only implies that these instances will be considered by the decision making process. Whether the agent fulfils or not these instances depends on its desires and its capabilities for achieving them.

7.5 Case Study

In the proposed case study, let us suppose that the *assistant* agent is not receiving the meteorological information and it is not able to determine if there is heavy rain or not. However, this instance was instantiated in the past (as described in Section 4.4.2.1)[10]. The coherence between the norm and the instance that has been created out of this norm is defined as the salience of the norm[11] as explained in Section 7.4.4. The coherence relationship among the belief that supports that the agent is the addressee of the instance and the instance itself is defined as a

[9] This agrees with principles 2a and 2c of deductive coherence

[10] The relevance of the instance (ρ_{NCC}) is 0.75

[11] The salience of the norm (ρ_{NAC}) is 0.35

half of the relevance of the instance (0.38)[12]. As being explained in Section 5.3, this instance is internalized and creates a new desire[13] to achieve the obligatory state. The coherence between the instance and the desire that has been created to fulfil the instance is 0.29 (the function for calculating coherence between desires and instances is explained in Section 7.4.4).

Moreover, there is a social norm that permits car drivers to maintain the speed when the road surface is dry:

$$norm(\langle \mathcal{P}, \neg slow(A), carDriver, surfaceDry(A), \neg surfaceDry(A), -, - \rangle, 0.1)$$

At some point in the past the *assistant* agent checks its visibility sensors and determines that the road is dry. Thus it had a belief as the following:

$$(\mathcal{B}\ surfaceDry(a_1), 1)$$

And the previous norm was instantiated as follows:

$$instance(\langle \mathcal{P}, \neg slow(a1), carDriver, self, surfaceDry(a_1), \neg surfaceDry(a_1), -, - \rangle, 1)$$

But now, the *assistant* agent cannot check its visibility sensors, due to the bad visibility conditions, and the belief $(\mathcal{B}\ surfaceDry(a_1), 1)$ is removed from the belief base. Since the *assistant* agent cannot determine if the road surface is not dry, it cannot consider the instance as expired and the instance stills being in the NCC.

Figure 7.2 illustrates the coherence graph that corresponds to this situation. The nodes of the graph represent those propositions that form the cognitive elements of the *assistant* agent. According to the normative constraints described in Section 7.4.2 there is an incoherence between the two conflicting instances. The degree of the inconsistency is calculated as follows:

$$min(0, 1 - 1 - 0.75) = -0.75$$

As Figure 7.2 shows, the partition that maximizes the coherence corresponds to the removal

[12] $\frac{\rho_{NCC}}{2}$

[13] $(\mathcal{D}\ slow(a_1), 0.29)$.

of the instance of the social norm. The coherence of this partition is 0.3[14]. The strength of the partition corresponding to the original graph is lower to the previous partition[15]. The strength of the partition corresponding to the removal of the obligation norm is also lower to the other two partitions[16]. As a consequence, the instance of the social norm is removed.

7.6 Contributions

The norm-base expansion process, which is described in Chapters 5 and 6, may cause conflicts with the cognitive elements of n-BDI agents; e.g., the internal motivations of agents (their desires). As illustrated in Section 2.6.1 existing proposals on norm-autonomous agents resolve conflicts by using static conflict resolution procedures. As a consequence, these proposals assume the existence of a priori preference ordering or utility function. However, the coherence maximization process carried out by the n-BDI proposal is able to "compute a realistic preference ordering considering the constraints that exist among the cognitive elements of an agent" [JcSSD10]. Therefore, coherence maximization can adapt to different personality traits depending on the cognitive elements present in the agent theory.

7.7 Conclusions

This chapter is focused on the coherence-based contraction mechanism that allows n-BDI agents to solve normative conflicts an conflicts that arise among norms and other mental propositions. Thanks to this mechanism, n-BDI agents are able to resolve mental conflicts dynamically by considering the cognitive and normative elements present in the agents' theory. In the next

[14] According to definition 7.2.3 the coherence of this partition is calculated as:

$$\sum_{(v,w)\in C_A} \frac{|\zeta(v,w)|}{|E|} = \sum_{(v,w)\in C_A} \frac{0.75+0.38+0.35+0.29}{6} = 0.3$$

[15] According to definition 7.2.3 the coherence of this partition is calculated as:

$$\sum_{(v,w)\in C_A} \frac{|\zeta(v,w)|}{|E|} = \sum_{(v,w)\in C_A} \frac{0.1+0.5+0.38+0.35+0.29}{6} = 0.27$$

[16] According to definition 7.2.3 the coherence of this partition is calculated as:

$$\sum_{(v,w)\in C_A} \frac{|\zeta(v,w)|}{|E|} = \sum_{(v,w)\in C_A} \frac{0.1+0.5+0.75}{6} = 0.23$$

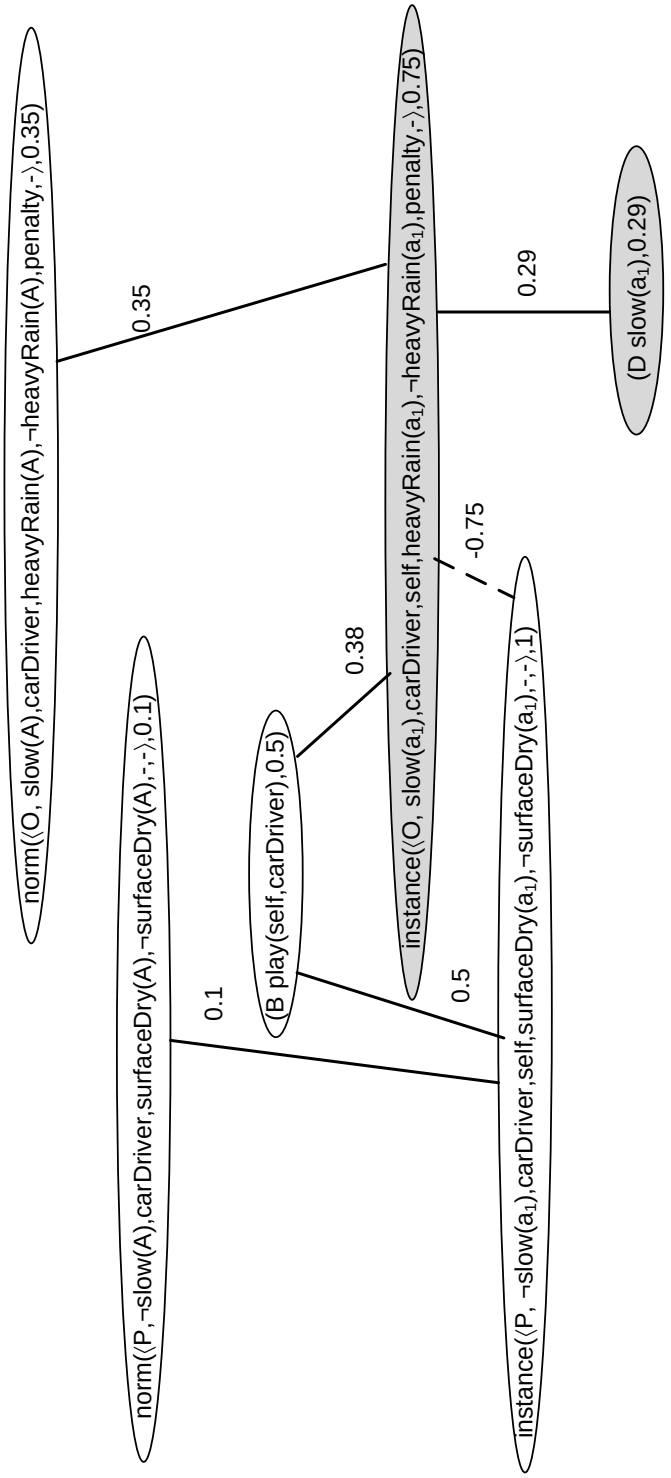

Figure 7.2: Coherence graph of the case study. Grey nodes correspond to the rejection set.

chapter the n-BDI architecture cold be applied into a fire-rescue case study.

Chapter 8

Case Study

The previous chapters have described the n-BDI Architecture and the reasoning process that n-BDI agents carry out. Along these chapters several brief case studies have illustrated the different aspects of the n-BDI architecture.

In the present chapter, we present a fire-rescue case study that allows us to evaluate if n-BDI agents achieve better results with respect to non-normative and norm-constrained agents in dynamic and complex environments. Specifically, we seek to determine whether the fact that agents can violate norms autonomously allow them to achieve a better adaptation to the environment.

8.1 Introduction

We consider two different types of persons: a fireman[1] and victims that must be rescued. Victims are located in a building in flames. Since they are not endowed with flame-proof clothes they wait until they are rescued by a fireman who leads victims to the door of the building. The fireman dies when there is not any path that allows him to reach the door. Therefore, the fireman decides to stop rescuing if he is taking too much risk.

There are norms that define general patterns that firemen must follow when dealing with fire threats. Specifically, we assume the existence of a norm that obliges firemen to abort the fire-rescue operation when it becomes too dangerous. However, there is a social norm that claims that foremen are permitted to violate the previous norm when a victim is on the verge of being reached.

[1]For simplicity, we assume that only one fireman participates in the fire-rescue operation.

This is a simple scenario in which there are norms in conflict. These norms may have different salience since the obligation norm is a formal norm, which has been explicitly defined by an authority, and the permission norm is a social norm, which has not been formally defined. Moreover, the circumstances in which these norms become relevant (i.e., risky situations and the probability of rescuing people) are uncertain. Finally, the environment (i.e., the building design and the position of victims) may change from fire-rescue to fire-rescue. Thus, decision making procedures that allow firemen to make decisions in these unforeseen fire-rescue scenarios are required.

8.1.1 Fire-Rescue Scenario Modelling

The fire-rescue case study has been modelled as a grid. Victims are randomly located in the grid. The fireman is initially located at the door of the building. For simplicity we have assumed that the building has one door. Initially there is one position on fire that is randomly distributed on the grid. In each iteration a new fire is randomly created on a free position of the grid. Figure 8.1 illustrates an example of a fire-rescue grid. This fire-rescue scenario is modelled as a grid of size 4, the door size is 3, and there are 3 victims that have not been rescued yet.

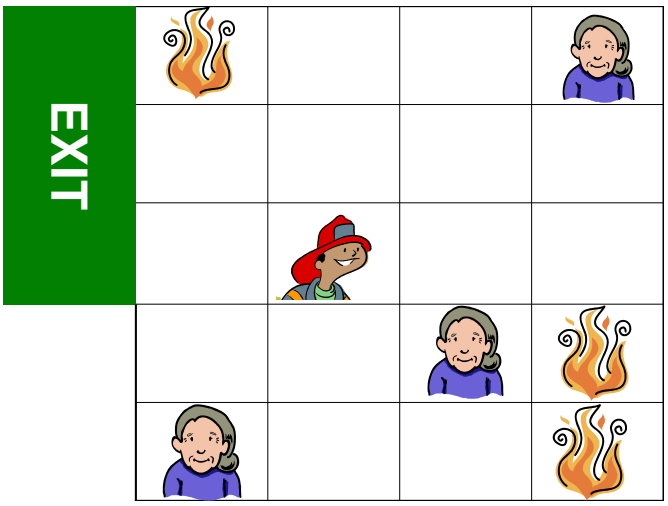

Figure 8.1: Example of a grid that models a building in flames

We have performed different simulations varying the size of the grid, the size of the door and the number of victims. In these simulations we compare the results that are obtained by three different implementations of the fireman: when the fireman does not consider norms,

when it implements crisis management norms as constraints, and when it is implemented as a n-BDI agent. Next, the different fireman implementations and the results obtained by these implementations are described in detail.

8.2 Non-Normative Fireman

In this implementation the fireman is not aware of norms. It moves randomly along the grid searching for victims. Java Function 8.1 contains the code that is executed by the fireman when it searches for victims in its surroundings.

Java Function 8.1: checkForVictimsInSurroundings Function

```
1  private List<Position> checkForVictimsInSurroundings(Position p) {
2    List<Position> victims=new ArrayList<Position>();
3    for(int i=-this.firemanPrecision;i<=this.firemanPrecision;i++){/*
       firemanPrecision determines the range of positions that can be observed
       */
4      for(int j=-this.firemanPrecision;j<=this.firemanPrecision;j++){
5        if(p.col+j>=0&&p.col+j<this.sizeGrid&&p.row+i>=0&&p.row+i<this.
           sizeGrid){
6          if(this.grid[p.row+i][p.col+j]==this.VICTIM){
7            victims.add(new Position(p.row+i,p.col+j));
8          }
9        }
10     }
11   }
12   return victims;
13 }
```

When the fireman finds a victim it tries to build a path to reach the victim. If this path exists, then the fireman tries to reach the victim. If the fireman is able to reach the victim, then it carries the victim to the door. Once the victim has been rescued, the fireman moves randomly again to find another victim. The fireman follows this pattern until either it rescues all victims that are reachable (i.e. that are not completely surrounded by fire) or it dies.

8.3 Norm-Constrained Fireman

In this implementation the fireman has not explicit knowledge about the norm that obliges it to abort a fire-rescue when it becomes too dangerous. On the contrary, the obligation norm has been implemented as a constraint as follows:

```
if(this.calculateFireManRisk()>=this.riskThreshold){
      this.exit=true; /*The fireman aborts the rescue*/
    }
```

where `this.calculateFireManRisk()` is a function that calculates the risk of a situation as a real number within the $[0, 1]$ interval; and `this.riskThreshold` is a real number within the $[0, 1]$ interval that represents the higher risk that the fireman takes. In each iteration, the fireman executes the risk function. If the value returned by this function is higher than the risk threshold, then the fireman stops the fire-rescue and it goes to the door.

Function 8.2 illustrates how the risk function has been implemented. Specifically, the risk of a given situation is calculated as the percentage of the surroundings that are on fire. For simplicity we have assumed that the fireman is able to determine whether the positions that are next to it are on fire or not.

Java Function 8.2: Risk Calculation Function

```
public double calculateFireManRisk(){
  int firedSurrounding=0;
  int surrounding=0;
  Position p=this.fireManPosition; /*fireManPosition contains the position
     of the fireman*/
  for(int i=-this.firemanPrecision;i<=this.firemanPrecision;i++){ /*
     firemanPrecision determines the range of positions that can be observed
     */
    for(int j=-this.firemanPrecision;j<=this.firemanPrecision;j++){
      if(p.col+j>=0&&p.col+j<this.sizeGrid&& p.row+i>=0&&p.row+i<this.
         sizeGrid){
        surrounding++;
        if(this.grid[p.row+i][p.col+j]==this.FIRE){
          firedSurrounding++;

        }
      }
    }
  }
  return (double)firedSurrounding/(double)surrounding;
}
```

8.4 n-BDI Fireman

The n-BDI fireman has explicit knowledge about the two norms that control the fire-rescue scenario. The fireman can take advantage of this to be able to violate the obligation norm.

The norm that obliges firemen to abort the fire-rescue when the situation becomes too risky is formally defined as:

$$\langle \mathcal{O}, abortRescue, fireman, risk, -, -, - \rangle^2$$

The permission norm that allows firemen to continue with the fire-rescue if they are about to rescue a victim is defined as:

$$\langle \mathcal{P}, \neg abortRescue, fireman, saveVictim, -, -, - \rangle$$

For simplicity, we assume that the fireman agent knows these two norms. Specifically, we assume that these two norms are equally salient (i.e., the salience of these two norms is 0.5). Therefore, according to the process described in Section 4.3, the NAC contains two propositions such as:

$$norm(\langle \mathcal{O}, abortRescue, fireman, risk, -, -, - \rangle, 0.5),$$
$$norm(\langle \mathcal{P}, \neg abortRescue, fireman, saveVictim, -, -, - \rangle, 0.5)$$

The obligation norm becomes relevant when there is a risky situation. The risk of a situation is also calculated by Function 8.2. The permission norm becomes effective when the fireman is able to save another victim. The probability of saving one more victim is calculated by Function 8.3. When the fireman is carrying a victim then the probability of saving this victim is 1. If it is not the case, the fireman looks its surroundings and searches for victims. The probability of saving these victims is calculated by considering the Manhattan distance [Kra75] between the positions of the fireman and the victim. `LinearFunction` is a function that returns the probability value, which decreases linearly as the distance increases.

Java Function 8.3: Victim Rescue Probability Function

```
1 private double probabilitySaveVictims() {
2   double prob=0.0;
3   if(this.fireManCarringVictim)  return 1.0; /* The fireman is carrying a
       victim*/
4   Position p=this.fireManPosition;
5   for(int i=-this.firemanPrecision;i<=this.firemanPrecision;i++){
6     for(int j=-this.firemanPrecision;j<=this.firemanPrecision;j++){
7       if(p.col+j>=0&&p.col+j<this.sizeGrid&& p.row+i>=0&&p.row+i<this.
```

[2]For simplicity we have assumed that once a risky situation is detected the norm is active. The norm has not expiration condition (i.e., it does not expire when the situation is not risky) and it only expires when the fireman leaves the building.

```
          sizeGrid){
8         if(this.grid[p.row+i][p.col+j]==this.VICTIM){
9           double newProb=LinearFunction(manhattanDistance(p,new Position(p.
              row+i,p.col+j)));
10          if(prob<newProb) prob=newProb;
11        }
12      }
13    }
14  }
15  return prob;
16 }
```

The process of norm compliance reasoning (i.e., determining if the obligation will be obeyed or not) is performed by Function 8.4. According to the Deontic Instance Activation Bridge Rule (see Equation 4.3 in Section 4.4.2.1) deontic norms are instantiated inside the agent's mind when their activation condition hold and the agent is under the influence of these norms. In this case study, the fireman believes that it is equally affected by the two norms. The influence of the permission (`this.permisionInfluence`) is equal to the influence of the obligation (`this.obligationInfluence`). The relevance of instances is calculated as a symmetric sum between the certainty about the activation of the norm and the certainty about the influence of the norm (lines 2-5 in Function 8.4). As defined by the normative constraints detailed in Section 8.3, norms that oblige to achieve something (i.e., *abortRescue*) and that permit to achieve it (i.e., $\neg$*abortRescue*) are in conflict when the sum of the relevance values of the two norms is higher than 1. In this case, there is a conflict and a coherence maximization process must be carried out. The two norms are equally salient (as previously mentioned both `this.obligationSalience` and `this.permissionSalience` are defined as 0.5). Thus, for simplicity we assume that if the relevance of the permission is higher than the relevance of the obligation (`relevanceP>relevanceO`), then the obligation is violated and the agent decides to continue with the fire-rescue operation (line 6 in Function 8.4). If it is not the case, then the fireman executes the Obligation Internalization Bridge Rule (see Equation 5.1 in Section 5.2.1). According to this bridge rule when the value calculated by the willingness function (`willinessObligation()`) is higher than the compliance threshold, a new desire is created for achieving the obliged condition. For simplicity, we assume that the compliance threshold is 0. However, the creation of the new desire does not imply that the obligation is fulfilled. The

new desire may be ignored if its desirability degree is low with respect to the degrees of the desires that are also contained in the DC. In this experiment we assumed that the obligation is ignored when the degree of the new desire is lower than the internalization threshold (i.e., `internalization>=this.internalizationThreshold`), see lines 7-10 in Function 8.4.

Java Function 8.4: Compliance With Obligation Function

```
1  private boolean complianceWithObligation() {
2    double risk=calculateFireManRisk();
3    double relevance0=symmetricSum(this.obligationInfluence,risk);
4    double prob=probabilitySaveVictims();
5    double relevanceP=symmetricSum(this.permisionInfluence,prob);
6    if(relevanceP+relevance0>=1.0 && relevanceP>relevance0) return false; /*
        The obligation is violated*/
7    double willingness0=willinessObligation();
8    double internalization=symmetricSum(relevance0,abs(willingness0));
9    if(willingness0>0.0&&internalization>=this.internalizationThreshold)
        return true; /*The obligation is fulfilled*/
10   return false; /*The obligation is ignored*/
11 }
```

According to the definition of the willingness function that is explained in Section 5.3, the willingness function is calculated as a weighted average among the three *willingness factors*: *self-interest* motivations, the *expectations* of being rewarded or sanctioned by others, and *emotional* factors. Function 8.5 contains the Java code corresponding to this functionality. In this implementation, we assume that the fireman does not want to abort the fire-rescue (`this.desAbort=0`). Since the obligation norm is not sanctioned, then the fireman has no expectation of being sanctioned (`this.desNegSanction=0`). Therefore, the weights of the willingness functions are defined as: `this.winterest=0`, `this.wexpectation=0`, `this.wemotion=1`. Thus, this case-study helps us to illustrate how the n-BDI architecture explains compliance with norms even if norms do not affect directly the agent goals. The indirect consequences of the obligation violation are that the fireman takes more risk and that more victims can be rescued. The concrete desirability of these prepositions (`this.desSurvive` and `this.desSaveVictims`) determines the personality of the fireman.

Java Function 8.5: Willingness To Comply With Obligation Function

```
1  private double willinessObligation() {
```

```
2   double interest=this.desAbort; /*Self-interest Factor*/
3   double expectation=this.desNegSanction; /*Expectation Factor*/
4   double emotion=this.obligationSalience+(((this.desSurvive*this.
        calculateFireManRisk())-(this.desSaveVictims*this.
        probabilitySaveVictims()))/(this.probabilitySaveVictims()+(1-this.
        calculateFireManRisk())));/*Emotional Factor*/
5   return ((this.winterest*interest)+(this.wexpectation*expectation)+(this.
        wemotion*emotion))/(this.winterest+this.wexpectation+this.wemotion);
6 }
```

8.5 Experimental Description

The main goal of the experiments that we performed is to determine whether the implementation of the fireman as a n-BDI agent improves its performance in a fire-rescue operation with respect to the results obtained by a non-normative and a norm-constrained fireman. To this aim, we performed simulations in which the different parameters of the grids (i.e., the size of the grid, the number of victims and the size of the door) are changed. After this, we compared the results obtained by the three implementations.

8.5.1 Metrics

There are two main factors that determine the success of a fire-rescue: the percentage of victims that are rescued and the survival of the fireman.

A simulation is represented as a set (G, D, V, R, F), where: G is the size of the grid; D is the door size; V is the total number of victims; R is the number of victims that have been rescued; and F takes value 1 when the fireman survives to the fire-rescue operation, otherwise it takes value 0.

The victim survival percentage achieved in a single simulation (G, D, V, R, F) is defined as:

$$s_V(G, D, V, R, F) = \frac{R}{M(G, D, V)}$$

where M is a function that returns the maximum number of victims that can be rescued on average for each grid size (G), door size (D) and number of victims (V)[3].

[3]To estimate the values returned by this function, we had previously performed a set of simulations using

Given a set of simulations ($\mathcal{N} = \{(G_1, D_1, V_1, R_1, F_i), ..., (G_N, D_N, V_N, R_N, F_n)\}$) the *victim survival percentage* (S_V) is defined as:

$$S_V(\mathcal{N}) = \frac{\sum_{i=1}^{N} s_V(G_i, D_i, V_i, R_i, F_i)}{N}$$

The *fireman survival percentage* (S_F) achieved in a set of simulations $\mathcal{N} = \{(G_1, D_1, V_1, R_1, F_i), ..., (G_N, D_N, V_N, R_N, F_N)\}$ is defined as:

$$S_F(\mathcal{N}) = \frac{\sum_{i=1}^{N} F_i}{N}$$

We define the *success* (S) of a set of simulations as the product between the values calculated by S_V and S_F for this set of simulations:

$$S(\mathcal{N}) = S_V(\mathcal{N}) * S_F(\mathcal{N})$$

8.5.2 Experiment Results

As previously mentioned, we performed different simulations for comparing the results obtained by the three implementations when the size of the grid, the size of the door and the number of victims change. Specifically, in each simulation the size of the grid (G) varies within the $[3, 15]$ interval, the size of the door (D) varies within the $[1, G]$ interval and the number of victims (V) varies within the $[1, \frac{(G-1)^2}{2}]$ interval. As previously mentioned, norm-constrained firemen use a risk threshold (`riskThreshold`). Similarly, n-BDI firemen use an internalization threshold (`internalizationThreshold`). The most suitable values for these thresholds depend on the characteristics of each grid. Therefore, it is not possible to determine a priori which is the best value for these thresholds. For this reason, in each simulation the two thresholds take random values within the $[0, 1]$ interval. For each value of G, D and V we performed 1000 different simulations to support the findings.

---

the non-normative fireman.

8.5.2.1 Average Results

This section illustrates the results that each fireman implementation obtains on average. Table 8.1 shows the victim survival percentage (S_V), the fireman survival percentage (S_F) and the success (S) that each implementation achieves in all the simulations.

Fireman Implementation	S_V	S_F	S
Non Normative	$99.98 \pm 20.02\%$	$19.54 \pm 9.95\%$	$19.53 \pm 16.96\%$
Norm-Constrained	$73.87 \pm 20.49\%$	$61.51 \pm 16.32\%$	$40.24 \pm 16.19\%$
Rational n-BDI	$64.98 \pm 0.05\%$	$93.27 \pm 0.19\%$	$59.52 \pm 0.19\%$
Coward n-BDI	$63.53 \pm 0.58\%$	$93.68 \pm 0.81\%$	$58.44 \pm 0.5\%$
Brave n-BDI	$71.21 \pm 0.49\%$	$85.6 \pm 0.26\%$	$58.87 \pm 0.41\%$

Table 8.1: 95% confidence interval for the victim survival percentage, the fireman survival percentage and the success that each implementation achieves in all the simulations.

As one could expect, non-normative firemen are able to rescue almost all the victims that can be rescued, since the firemen do not abort the fire-rescue ever. However, the firemen survival is very low. Therefore, the lowest success is obtained by non-normative firemen.

In case of norm-constrained firemen, they achieve better results since the firemen survival percentage is significantly higher, whereas the victim survival percentage decreases to a lesser extent. The confidence intervals in case of norm-constrained firemen are the largest. Hence, the behaviour of norm-constrained firemen is more uncertain and depends on the concrete value that the risk threshold takes.

As previously mentioned, different personalities of the n-BDI fireman can be implemented according to the values that `this.desSurvive` and `this.desSaveVictims` take. We have performed simulations considering three personalities: rational, coward and brave. As depicted in Table 8.1, the confidence intervals are the smallest in n-BDI firemen. Hence, the behaviour of n-BDI firemen is less dependent on the value of the compliance threshold. Thus, n-BDI firemen are able to adapt better to different configurations of the fire-rescue grid. *Rational* firemen are those ones that want to preserve victims' life as much as they want to preserve its own life. Therefore, both `desSurvive` and `desSaveVictims` have been set to 1. As the results show, rational firemen are more cautious than norm-constrained firemen and their survival (S_{GF}) is much higher. As a result, the victim survival percentage (S_V) decreases. However, it decreases to a lesser extent and a higher success is obtained by rational firemen. *Coward* firemen are those ones that want to preserve victims' life less than they want to preserve its own life. Therefore,

`desSurvive=1` and `desSaveVictims=0.5`. As a consequence, the fireman survival percentage (S_F) increases. Since coward fireman takes less risks, then the number of rescued victims decreases. As a consequence, the success that is obtained by coward firemen is lower than rational firemen. Finally, *brave* firemen are those ones that want to preserve victims' life more than they want to preserve their own life. Therefore, `desSurvive=0.5` and `desSaveVictims=1`. As a consequence, the victim survival percentage (S_V) increases notably. On the contrary, brave firemen take more risks and their survival decreases lightly. As a consequence, the success obtained by brave firemen is higher than the other n-BDI firemen.

In light of these results, we can conclude that norms help the fireman to achieve better results. In the two implementations that consider norms (i.e., the norm-constrained and the n-BDI implementation) the fireman survival percentage is higher. Specifically, the fireman survival percentage is the highest in case of the n-BDI fireman. We can conclude that the n-BDI implementation achieves a higher success on average.

8.5.2.2 Detailed Results

As previously mentioned, in the simulations performed the values of the risk and compliance thresholds vary randomly within the $[0, 1]$ interval. To illustrate the performance of the norm-constrained and n-BDI implementations in different situations we analysed the results that each type of fireman obtains according to the value of the thresholds. Specifically, we have classified the simulations in three categories: *low thresholds*, when both `riskThreshold` and `internalizationThreshold` vary within the $[0, 0.33)$ interval; *medium thresholds*, when both `riskThreshold` and `internalizationThreshold` vary within the $[0.33, 0.66)$ interval; and *high thresholds*, when both `riskThreshold` and `internalizationThreshold` vary within the $[0.66, 1]$ interval. Next, the results obtained for each category are described.

Low Thresholds. This section details the results obtained when the thresholds take low values. Notice that a low value of the `riskThreshold` means that norm-constrained firemen take less risks and less victims are rescued. Similarly, a low value of the `internalizationThreshold` means that n-BDI firemen are more prune to follow the rescue-abandoning norm and less victims are rescued. The results obtained in these simulations are shown in Table 8.2.

In all cases, the fireman survival percentage (S_F) is higher than the average values. This

Fireman Implementation	S_V	S_F	S
Norm-Constrained	$47.05 \pm 21.01\%$	$96.79 \pm 5.89\%$	$44.79 \pm 19.21\%$
Rational n-BDI	$44.82 \pm 0.08\%$	$98.08 \pm 0.31\%$	$43.58 \pm 0.31\%$
Coward n-BDI	$41.26 \pm 0.97\%$	$98.4 \pm 0.31\%$	$40.3 \pm 0.88\%$
Brave n-BDI	$49.5 \pm 0.85\%$	$97.27 \pm 0.21\%$	$47.56 \pm 0.8\%$

Table 8.2: 95% confidence interval for the victim survival percentage, the fireman survival percentage and the success when `riskThreshold` and `internalizationThreshold` vary within the $[0, 0.33)$ interval.

is caused by the low value of the thresholds that make firemen to take less risks. Due to the same reason, the victim survival percentage (S_V) decreases.

In case of the norm-constrained fireman, better results are obtained when the risk threshold take low values. In case of the n-BDI implementations, the success achieved by the three fireman personalities is lower than on average. This is caused by the fact that the fireman survival percentage (S_F) lightly increases whereas the victim survival percentage (S_V) decreases to a greater extent. Thus, only brave firemen achieve better results than norm-constrained firemen.

Medium Threshold. This section details the results obtained when the thresholds take medium values. The results obtained in these simulations are shown in Table 8.3.

Fireman Implementation	S_V	S_F	S
Norm-Constrained	$85.27 \pm 9.14\%$	$63.5 \pm 12.93\%$	$53.36 \pm 8.22\%$
Rational n-BDI	$71.12 \pm 0.08\%$	$93.39 \pm 0.32\%$	$65.78 \pm 0.32\%$
Coward n-BDI	$70.3 \pm 0.31\%$	$93.97 \pm 0.92\%$	$65.51 \pm 0.7\%$
Brave n-BDI	$77.88 \pm 0.45\%$	$88.12 \pm 0.42\%$	$67.81 \pm 0.34\%$

Table 8.3: 95% confidence interval for the victim survival percentage, the fireman survival percentage and the success when `riskThreshold` and `internalizationThreshold` vary within the $[0.33, 0.66)$ interval.

In all cases, the fireman survival percentage (S_F) is lightly higher than on average. Moreover, the victim survival percentage (S_V) is much higher than on average. Therefore, in all cases the success is higher than on average.

High Threshold. This section details the results obtained when the thresholds take high values. Notice that a low value of the `riskThreshold` means that norm-constrained firemen take more risks and more victims are rescued. Similarly, a low value of the `internalization-`

`Threshold` means that n-BDI firemen are less prune to follow the rescue-abandoning norm and more victims are rescued. The results obtained in these simulations are shown in Table 8.4.

Fireman Implementation	$\mathbf{S_V}$	$\mathbf{S_F}$	S
Norm-Constrained	$87.98 \pm 6.82\%$	$26.57 \pm 16.34\%$	$23.34 \pm 12.21\%$
Rational n-BDI	$77.93 \pm 0.08\%$	$88.66 \pm 0.34\%$	$68.4 \pm 0.34\%$
Coward n-BDI	$77.84 \pm 0.28\%$	$88.98 \pm 0.39\%$	$68.6 \pm 0.34\%$
Brave n-BDI	$85.09 \pm 0.35\%$	$72.24 \pm 0.54\%$	$60.84 \pm 0.34\%$

Table 8.4: 95% confidence interval for the victim survival percentage, the fireman survival percentage and the success when `riskThreshold` and `internalizationThreshold` vary within the $[0.66, 1]$ interval.

In all cases, the fireman survival percentage (S_F) is lower than on average. This is caused by the high value of the thresholds that make firemen to take more risks. Due to the same reason, the victim survival percentage (S_V) increases.

In case of the norm-constrained implementation, the high value of the risk threshold causes firemen take more risk even if they are not close to save a victim. For this reason, the decrease of the fireman survival percentage is higher than the increase of the victim survival percentage. As a result, the success achieved by norm-constrained fireman is lower than on average.

In case of n-BDI firemen, they achieve better results since they only take more risk when they believe that a victim can be rescued. This makes that the victim survival percentage (S_V) increases notably whereas the fireman survival percentage (S_F) decreases lightly. An interesting result if that brave firemen achieve worse results than the other n-BDI firemen. This is due to the fact that brave fireman take more risk since they consider victim's life as more important. This together with the high thresholds made that breve firemen take risks even if they are not really close to same a victim. This makes the fireman survival percentage (S_F) decreases more than the increase of the victim survival percentage (S_V).

8.6 Conclusions

In this chapter, several simulations of a fire-rescue case study have been developed to evaluate the n-BDI architecture. Specifically, we have modelled a fire-rescue case study following three different approaches: ignoring norms, implementing norms as constraints on agents and using the n-BDI architecture to implement firemen agents.

As the experimental results illustrate, the use of the n-BDI architecture allows us to model a more dynamic behaviour. The fact that agents can violate norms autonomously allows them to better adapt to their environment. Specifically, we have demonstrated that n-BDI agents are capable of self-adjusting their behaviour to the features of the fire-rescue operation in which they are involved. Moreover, different agent personalities can be modelled. Thereby, the behaviour of n-BDI agents is predictable to some degree and MAS designers can decide the behaviour of the agents according to the functionality that they require.

Chapter 9

MaNEA: A Distributed Architecture for Enforcing Norms in Open MAS

Norms have been promoted as a coordination mechanism for controlling agent behaviours in open MAS. Thus, agent platforms must provide normative support, allowing both norm-aware and non norm-aware agents to take part in MAS that are controlled by norms. Existing proposals present several drawbacks that make them unsuitable for open MAS. In response to these problems, this chapter describes a new Norm-Enforcing Architecture aimed at controlling open MAS, named MaNEA.

This chapter is organized as follows: Section 9.1 contains a short introduction to this chapter; Section 9.2 contains the analysis of the main proposals on infrastructural norm enforcement; Section 9.3 describes briefly the Magentix2 platform; Section 9.4 describes the main components of MaNEA; Section 9.6 illustrates the performance of MaNEA through a case study; Section 9.5 describes an implementation of a prototype of the n-BDI architecture in the Magentix2 platform; Section 9.7 contains an evaluation of this architecture; Section 9.8 summarises the main contributions of this chapter; and, finally, Section 9.9 contains a short conclusions.

9.1 Introduction

One of the main applications of MAS is its usage for supporting large scale open distributed systems. These systems are characterized by the heterogeneity of their participants; their limited trust; a high uncertainty; and the existence of individual goals that might be in conflict [AP01]. In these scenarios, norms are conceived as an effective mechanism for achieving co-

ordination and ensuring social order; i.e., norms represent an effective tool for regulating the actions of software agents and the interactions among them [LyLLd06]. Most of the proposals on methodologies and guidelines aimed at developing open MAS [ABJ11, DVSD05] are based on organizational concepts, such as norms. These concepts facilitate the analysis and design of coordination and collaboration mechanisms for MAS. Therefore, norms should be considered in the design and specification of the MAS [CAB11c]. As pointed out in [Cas03], the use of norms in MAS allows better results to be achieved in dynamic and complex environments. Agent platforms are the software that supports the development and execution of MAS. Thus, norms must be also considered in the design and implementation of agent platforms [CAB11c]. As a consequence, agent platforms must implement norms in an optimized way, given that in open MAS the internal states of agents are not accessible [CAB11a]. Therefore, norms cannot be imposed as agent's beliefs or goals, but they must be implemented in the platforms by means of *control* mechanisms [GAD07].

This chapter considers the main challenges of open MAS and points out the main deficiencies and drawbacks of agent platforms and infrastructures when supporting norms. With the aim of overcoming some of these problems, in this chapter a Norm-Enforcing Architecture, known as MaNEA, is proposed. Specifically, MaNEA has been integrated into the Magentix2 platform[1]. The Magentix2 platform allows the management of open MAS in a secure and optimized way. Its main objective is to bring agent technology to real domains: business, industry, e-commerce, among others. This goal entails the development of more robust and efficient mechanisms for enforcing norms that control these complex applications.

9.2 Related Work

Most of the proposals on norms for controlling MAS tackle this issue from a theoretical perspective [BvdT04a, Ser98]. However, there are also works on norms from a computational point of view. These works proposals have been described in Section 2.5.3. In this section we will provide a more complete overview of infrastructural norm enforcement.

[1] http://magentix2.gti-ia.upv.es/

9.2.1 Infrastructural Observability

Normative agent platforms provide entities that are in charge of both observing and enforcing norms. Cardoso & Oliveira [CO07] propose a norm-enforcing architecture in which the monitoring and enforcement of norms is made by a single institutional entity, named as *normative environment*. This entity receives all messages that have been exchanged among agents and determines if an agent has violated (vs. fulfilled) a norm. In this case, the *normative environment* sends a sanctioning (vs. rewarding) notification to this agent. As argued by Cardoso & Oliveira the implementation of the *normative environment* as a centralized component represents a performance limitation when dealing with a considerable number of agents.

To address the performance limitation of centralized approaches, distributed mechanisms for an institutional enforcement of norms are proposed in [MU00, GGCN+07]. These works propose languages for expressing norms and software architectures for the distributed enforcement of these norms. In [MU00], Minsky & Ungureanu present an enforcement mechanism that is implemented by the Moses toolkit [MU98]. Its performance is as general (i.e., it can implement all norms that are controllable by a centralised enforcement) and more scalable and efficient than centralized approaches. However, one of the main drawbacks of this proposal is the fact that norms can only be expressed in terms of the messages sent or received by an agent; i.e., this framework does not support the definition of norms that affect an agent as a consequence of an action carried out independently by another agent. This problem is overcome by Gaertner et al. in [GGCN+07]. In their approach, Gaertner et al. propose a distributed architecture for enforcing norms in EI. Specifically, this architecture only controls dialogical actions. Thus, the dialogical actions performed by agents cause the propagation of normative propositions (i.e., obligations, permissions, and prohibitions). These normative propositions are taken into account by the normative level; i.e., a higher level in which norm reasoning and management processes are performed in a distributed manner.

In a more recent work, Modgil et al. propose in [MFM+09] a general architecture for monitoring norm-governed systems. Specifically, it is a two layer architecture in which observers (i.e., the lowest layer) are capable of reporting to monitors (i.e., the highest layer) on states of interest relevant to the activation, fulfilment, violation and expiration of norms. Monitors determine if a violation or fulfilment has occurred and they take remedial actions accordingly. The proposal of Modgil et al. does not give any detail of how the monitoring and observation levels can be dynamically distributed into a set of coordinated entities in response to a changing

environment. Thus, this architecture is not capable of dynamically adapting to situations in which the number of norms and agents to be controlled may change drastically.

9.2.2 Requirements for Norm Enforcing Architectures

As being illustrated by the previous section, infrastructures that provide support to norm-enforcing present some drawbacks that make them unsuitable for controlling norms in open MAS. In summary, the most important requirements for norm-enforcing architectures are:

- **Automatic Enforcement.** It must provide support for the detection of norm violations and the application of remedial mechanisms. It implies that agents can trust the enforcement system that will sanction their partners if they behave dishonestly. Moreover, the enforcement architecture must provide normative information in order to allow agents to realise that they or other agents have violated a norm. Thus, agents are persuaded to obey norms not only by a material system of sanctions but also since their non-normative behaviour can be observed by others, who may reject to interact with them in the future.

- **Control of general norms.** It must control complex and general norms. Thus, it must allow the definition and management of norms that control not only the messages exchanged among agents but also other actions carried out by agents. In addition, it must support the enforcement of norms that control states of affairs. Finally, it must control norms that are defined in terms of actions and states of affairs that occur independently (e.g., actions that are performed by different agents).

- **Dynamic Enforceable Content**. Dynamic situations may cause norms to lose their validity or to need to be adapted. Thus, norm-enforcing mechanisms should provide solutions to open MAS in which the set of norms evolves along time. Moreover, it must provide support for the enforcement of unforeseen norms that control activities and actions that are defined on-line.

- **Efficient, Distributed and Robust**. Finally, enforcement mechanisms must bring the possibility of performing this task in a distributed way. This distributed architecture must be capable of operating quickly, effectively and orderly in changing environments in which the number of agent, norms etc. may change drastically.

Table 9.1 summarizes the performance of the proposals on infrastructural enforcement with respect to these requirements. In particular, the *automatic enforcement* feature consists on three different activities: (i) the detection of norm violations (*Violation Detection* column of Table 9.1), the application of sanctions and rewards (*Remedial Application* column) and the provision of normative information (*Normative Information* column). With regard to the type of norms that these proposals control, they have been evaluated according to four criteria: the possibility of controlling the messages exchanged by agents (*Message Exchange* column), the possibility of controlling actions performed by agents (*Action Performance* column), the possibility of controlling states of affairs (*States of Affairs* column) and the possibility of controlling norms that affect an agent due to a certain action or message sent by other agent (*Independent Situations* column). The suitability of these infrastructures for controlling dynamic environments has been evaluated according to two criteria: the consideration of norms that are only active under specific circumstances (*Norm Evolution* column) and the explicit consideration of norm change (i.e., creating and deleting norms on-line) (*Norm Modification* column). Finally these proposals have been evaluated according to the possibility of distributing the norm enforcing architecture (*Distributed Architecture* column). As illustrated in this table, issues such as the provision of *normative information* and the explicit consideration of the *norm modification* problem have not been properly addressed by the existing proposals. With the aim of meeting these pending requirements and improving the efficiency of existing approaches in terms of the messages that are required to control norms, we propose in Section 9.4 a Norm-Enforcing Architecture for controlling norms in the Magentix2 platform. Specifically, the Norm-Enforcing Architecture bases on the organization and interaction support offered by Magentix2. Next, the Magentix2 platform is briefly described.

9.3 The Magentix2 Platform

Magentix2 is an agent platform for open MAS in which heterogeneous agents interact and organize themselves into VOs [FKT01]. VOs are open systems formed by the grouping and collaboration among heterogeneous entities and there is a separation between form and function that requires defining how behaviour will take place [FMB05]. VOs are social entities formed by agents that try to achieve the organizational goals. These agents are organized in groups that are controlled by norms.

	Violation Detection	Remedial Application	Normative Information	Message Exchange	Action Performance	States of Affairs	Independent Situations	Norm Evolution	Norm Modification	Distributed Architecture
Cardoso & Oliveira [CO07]	√	√	-	√	-	-	√	-	-	-
Minsky & Ungureanu [MU00]	√	√	-	√	-	-	-	-	-	√
Gaertner, et al. [GGCN+07]	√	√	-	√	√	-	√	√	-	√
Modgil, et al. [MFM+09]	√	√	-	√	√	√	√	√	-	√

Table 9.1: Summary of distributed proposals on infrastructural enforcement

Magentix2 provides support for VOs at two levels:

- *Organization level.* Magentix2 provides access to the organizational infrastructure [ABC+11] through a set of services included on two main components: the *Service Facilitator* [dVCR+09], which is a service manager that registers the services provided by entities and facilitates service discovering for potential clients; and the *Organization Management System* (OMS) [CJBA10], which is in charge of the management of VOs, taking control of their underlying structure, the roles played by agents, and the register of the norms that govern the VO.

- *Interaction level.* Magentix2 provides support to: *agent communication*, supporting asynchronous reliable message exchanges and facilitating the interoperability between heterogeneous entities; *agent conversations* [FAS+10], which are automated Interaction Protocols; *tracing service support* [BGFJT11], which allows agents in a MAS to share information in an indirect way by means of trace events; and, finally, Magentix2 incorporates a *security module* [SEGFB11] that provides features regarding security, privacy, openness and interoperability.

Norms define what is considered as permitted, forbidden or obligatory in an abstract way. However, norm compliance must be controlled considering the actions and messages exchanged

among agents at the interaction level. The Norm-Enforcing Architecture proposed in this chapter tries to fill the gap between the organizational level, at which norms are registered by the OMS; and the interaction level, at which actions and communications between agents can be traced. Next, the Tracing Service Support and the storage of norms, provided by the OMS, are described.

9.3.1 Tracing Service Support

In order to facilitate indirect communication (i.e., indirect ways of interaction and coordination), Magentix2 provides Tracing Service Support [BGFJT11]. This service is based on the publish/subscribe software pattern, which allows subscribers to filter events attending to some attributes (content-based filtering), so that agents only receive the information in which they are interested and only requested information is transmitted. In addition, security policies define which entities are authorized to receive which specific events. These tracing facilities are provided by a set of components named Trace Manager (TM). There can be three types of *tracing entities* (i.e., those elements of the system capable of generating and/or receiving events): agents, artifacts or aggregations of agents.

A trace event or *event* is a piece of data representing an action, message exchange or situation that has taken place during the execution of an agent or any other component of the MAS. *Generic* events, which represent application independent information, are *instrumented* within the code of the platform. *Application* events are domain dependent information.

Definition 9.3.1 (Event) *An event e is defined as a tuple $e = \langle Type, Time, Origin, Data \rangle$, where:*

- *$Type$ is a constant that represents the nature of the information represented by the event;*
- *$Time$ is a numeric value that indicates the global time at which the event is generated;*
- *$Origin$ is a constant that identifies the tracing entity that generates the event;*
- *$Data = \psi_1 \wedge ... \wedge \psi_n$ is a conjunction of possibly negated first-order grounded atomic formulae that contains extra attached data required for interpreting the event.*

Trace events can be processed or even combined to generate compound trace events, which can be used to represent more complex information.

Any tracing entity is provided with mail boxes for receiving or delivering events (E_{In} and E_{out}). Entities that want to receive certain trace events request the subscription to these events by sending to the TM a *subscription* event that contains the template of those events they are interested in.

Definition 9.3.2 (Template) *A template t is a tuple $t = \langle Type, Origin, Data \rangle$ that contains the filtering specified criteria for events, where:*

- *$Type$ is a constant that represents the nature of the information represented by the event;*
- *$Origin$ is a constant that identifies the entity that generates the event;*
- *$Data = \psi_1 \wedge ... \wedge \psi_n$ is a conjunction of possibly negated first-order atomic formulae that may contain free variables.*

Let us consider the standard notion of substitution as a finite and possibly empty set of pairs X/y where X is a variable and y is a term. Let us also define the application of a substitution σ as:

Phase 1. $\sigma(c) = c$ if c is a constant.

Phase 2. $\sigma(X) = y$ if $X/y \in \sigma$; otherwise $\sigma(X) = X$.

Phase 3. $\sigma(\psi_1 \wedge ... \wedge \psi_n) = \sigma(\psi_1) \wedge ... \wedge \sigma(\psi_n)$.

Phase 4. $\sigma(\langle \rho_0, ..., \rho_n \rangle) = \langle \sigma(\rho_0), ..., \sigma(\rho_n) \rangle$

Therefore, the application of a substitution on a template is defined as follows:

$$\sigma(\langle Type, Origin, Data \rangle) = \langle Type, Origin, \sigma(Data) \rangle$$

since $Type$ and $Origin$ take constant values.

According to the definitions of events and templates the *matching* relationship between events and templates is defined as follows:

Definition 9.3.3 (Matching Function) *Given an event $e = \langle Type, Time, Origin, Data \rangle$ and a template $t = \langle Type', Origin', Data' \rangle$, their matching is a boolean function defined as follows:*

$$matching(e,t) = \begin{cases} true & if\ (Type = Type') \wedge \\ & ((Origin = Origin') \\ & \vee (Origin'\ is\ undefined)) \\ & \wedge (\forall \psi_i \in Data' : \psi_i \in Data) \\ false & otherwise \end{cases}$$

Definition 9.3.4 (Unification Function) *Given an event e and a template t, their unification is a boolean function defined as follows:*

$$unification(e,t) = \begin{cases} true & if\ exists\ a\ substitution \\ & of\ variables\ \sigma\ such\ that \\ & matching(e, \sigma(t))\ is\ true \\ false & otherwise \end{cases}$$

9.3.2 Organization Management System (OMS)

The Organization Management System (OMS) [CJBA10] is responsible for the management of VOs and their constituent entities. The OMS provides a set of services: **structural services**, which comprise services for adding/deleting norms (*registerNorm* and *deregisterNorm* services allow entities to modify the norms that are in *force* or applicable within a VO), and for adding/deleting roles and groups; **informative services**, that provide information of the current state of the organization; and **dynamic services**, which allow agents to enact/leave roles inside VOs (*acquireRole* and *leaveRole* services). Moreover, agents can be forced to leave a specific role (*expulse* service). When the OMS provides any of these services successfully, then it generates an event for informing about the changes produced in the VO.

9.3.2.1 Norm Definition

According to the normative definitions provided in Chapter 3, in Magentix2 a distinction among *norms* and *instances* is made. This chapter only considers deontic norms (see Definition 3.2.1 in Section 3.2). Thus, we will use the term norm as a synonym of deontic norm. Magentix2 takes a closed world assumption where everything is considered as permitted by default. Therefore, permissions are not considered in this chapter, since they can be defined as normative operators

that invalidate the activation of an obligation or prohibition. For the purpose of this chapter, we redefine norms as follows:

Definition 9.3.5 (Norm) *A norm (n) is defined as a tuple* $n = \langle id, D, T, A, E, C, S, R \rangle$*, where:*

- *id is the norm identifier;*
- $D \in \{\mathcal{F}, \mathcal{O}\}$ *is the deontic modality of the norm,* $\mathcal{F}$ *represents prohibition and* $\mathcal{O}$ *represents obligation;*
- T *is the target of the norm, the role to which the norm is addressed;*
- A *is the norm activation condition, it defines under which circumstances the norm is active and must be instantiated;*
- E *is the norm expiration condition that determines when the norm expires and no longer affects agents;*
- C *is the norm condition that represents the action or state of affairs that is forbidden or obliged;*
- S *and* R *describe the sanctioning and rewarding actions that will be carried out in case of norm violation or fulfilment, respectively.*

As previously argued, MaNEA builds on the event tracing approach to monitoring. Thus, the conditions A, E and C are expressed in terms of event templates.

In Magentix2 norms can be classified into two main categories: organizational and functional norms. Examples of norms belonging to each category are provided below.

Organizational Norms Organizational norms [CJBA10] are related to services offered by the OMS to members of the organization. They establish organizational dynamics, e.g. role management (role cardinalities, incompatibility between roles) and the protocol by which agents are enabled to acquire roles. For example, an organizational norm that forbids any agent to register new norms when there is a critical situation is defined as follows:

$$\langle n_1, \mathcal{F}, member,$$
$$\langle restrictedNormativeChange, -, - \rangle, \langle freeNormativeChange, -, - \rangle,$$
$$\langle registerNorm, -, norm(N) \rangle, -, - \rangle$$

According to norm n_1 once the $restrictedNormativeChange$ event is sent, any agent that enacts the *member* role (it is a special role that is implicitly played by all agents in Magentix) is forbidden to request the register of any norm (i.e., any event the matches the template $\langle registerNorm, -, norm(N)\rangle$ will be considered as forbidden). This norm will remain active until the $freeNormativeChange$ event is received.

Similarly, an incompatibility constraint between two roles (r_1 and r_2), which define that agents cannot play simultaneously roles r_1 and r_2, is modelled by the two following norms:

$$\langle n_2, \mathcal{F}, r_1,$$
$$\langle incompatibilityActivation, -, -\rangle, \langle incompatibilityExpiration, -, -\rangle,$$
$$\langle aquireRole, -, role(r_2)\rangle, -, -\rangle$$

$$\langle n_3, \mathcal{F}, r_2,$$
$$\langle incompatibilityActivation, -, -\rangle, \langle incompatibilityExpiration, -, -\rangle,$$
$$\langle aquireRole, -, role(r_1)\rangle, -, -\rangle$$

For example, norm n_3 defines that once the $incompatibilityActivation$ event has been sent, then any agent that enacts role r_1 is forbidden to request the acquisition of role r_2. This norm will remain active since the $incompatibilityExpiration$ event is sent.

Functional Norms Functional norms [CJBA10] are domain dependent norms that define the functionality of roles. For example, let us suppose the case of an assembly line that has been implemented as a hierarchy of agents. Thus, there is a set of robots that perform the different assembly tasks; i.e., the *subordinated*. These robots are controlled by a set of agents that monitor and evaluate their performance; i.e., the *supervisors*. Supervisors are responsible for dynamically reorganizing robots in the assemble line to improve the productivity. To this aim, robot agents are asked for reporting information about their performance to *auditor* agents that will analyse the performance of the assembly line. This situation can be modelled as a functional norm defined as follows:

$$\langle n_4, \mathcal{O}, subordinated,$$
$$\langle auditStart, -, task(T)\rangle, \langle auditEnd, -, task(T)\rangle,$$
$$\langle taskPerformance, -, performance(P) \wedge task(T)\rangle,$$
$$-, -\rangle$$

When an audit stage of a given task (T) starts (i.e., the *auditStart* event is sent), *subordinated* agents are obliged to inform about their performance on this task before the audit stage ends (i.e., the *auditEnd* event is sent).

9.3.2.2 Instance Definition

As mentioned in Section 3.2, when the activation condition of a norm holds; i.e., the activation event is detected, then it becomes active and several *instances* are created, according to the possible groundings of the activation condition. For the purpose of this chapter, we redefine instances as follows:

Definition 9.3.6 (Instance) *Given a norm $n = \langle id, D, T, A, E, C, S, R\rangle$ and a perceived event e, an instance i of n is the tuple $i = \langle id', D', T', E', C', S', R'\rangle$, where:*

- $unification(e, A)$ *is true, i.e., there is a substitution σ such that $matching(e, \sigma(A))$ is true (the norm is active);*
- $C' = \sigma(C)$, $E' = \sigma(E)$, $S' = \sigma(S)$, *and* $R' = \sigma(R)$;
- $id' = id, D' = D$ *and* $T' = T$.

For example, let us suppose that the event

$$\langle auditStart, t, s_1, task(assembling)\rangle$$

is sent by an agent (s_1). Thus, norm n_4 will be instantiated as follows:

$$i_{assembling} = \langle n_4, \mathcal{O}, subordinated,$$
$$\langle auditStart, -, task(assembling)\rangle, \langle auditEnd, -, task(assembling)\rangle,$$
$$\langle taskPerformance, -,$$
$$performance(P) \wedge task(assembling)\rangle, -, -\rangle$$

Definition 9.3.7 (Instantiation Function) *Given an event $e = \langle Type, Time, Origin, Data\rangle$ and a norm $n = \langle id, D, T, A, E, C, S, R\rangle$, instantiation is a function that instantiates norm n as follows:*

$$instantiation(e, n) = \langle id', D', T', E', C', S', R'\rangle$$

where

- *there is a substitution* σ *such that* $matching(e, \sigma(A))$ *is true;*
- $C' = \sigma(C)$, $E' = \sigma(E)$, $S' = \sigma(S)$, *and* $R' = \sigma(R)$*;*
- $id' = id, D' = D$ *and* $T' = T$*.*

9.3.2.3 Power Definition

Once the norm activation event has been detected and a new instance is created, all agents playing the target role are under the influence of the new instance. Thus, a normative *power* (or power for short) represents the control over a concrete agent that is playing the target role.

Definition 9.3.8 (Power) *Given an instance* $i = \langle id', D', T', E', C', S', R' \rangle$*, a power* p *is a tuple* $p = \langle id'', D'', T'', C'', S'', R'', W'' \rangle$ *where:*

- $id'' = id', D'' = D', T'' = T', S'' = S', R'' = R'$ *are defined as in the instance;*
- $C'' = \langle C'_{Type}, AgentID, C'_{Data} \rangle$ *such that* $C' = \langle C'_{Type}, -, C'_{Data} \rangle$ *and* $AgentID$ *is a constant that identifies the agent affected by the power;*
- W'' *is a boolean constant that expresses if the event* C'' *has been received.*

For example, let us suppose that there is a robot agent r_1 that is playing the *subordinated* role. Thus, a new power for controlling the behaviour of r_1 according to n_4 will be created as follows:

$$p_{r_1,assembling} = \langle n_3, \mathcal{O}, subordinated, \langle taskPerformance, r_1, performance(P) \wedge task(assembling) \rangle, -, -, false \rangle$$

The next section describes the Norm-Enforcing Architecture proposed in this chapter. It is a two layer architecture formed by: a higher level in charge of detecting the instantiation of norms; and a lower level in charge of enforcing powers on agents. The operational semantics of norms, instances and powers (i.e., how they are created, deleted, fulfilled and violated) is explained below.

9.4 Norm-Enforcing Architecture: MaNEA

The main purpose of MaNEA (*Magentix2 Norm-Enforcing Architecture*) is to endow the Magentix2 platform with an infrastructure capable of controlling norms in open MAS where unforeseen scenarios may occur. Therefore, the number of agents and the situations that must be controlled through norms may change at runtime. For this reason, MaNEA has been designed as a distributed architecture. Specifically, MaNEA has been distributed into two layers. The highest layer is formed by *Norm Manager* (NM) entities that control all processes related to the creation and elimination of both norms and instances. The lowest layer is formed by *Norm Enforcer* (NE) entities that are responsible for controlling the agents' behaviours.

9.4.1 Norm Manager

The Norm Manager (NM) is responsible for determining which norms are active (i.e., have to be instantiated) at a given moment. Algorithm 1 illustrates the pseudocode of the control loop performed by the NM. When the NM receives an event (e), then it handles the event according to the event type. Mainly, the NM carries out a process that can be divided into two differentiated tasks: norm management and instance management. Thus, the NM maintains a list (N) that contains all norms that have been registered in Magentix2 and a list (I) that contains all instances that remain active at a given moment.

9.4.1.1 Norm Management

In order to maintain the norm list, the NM subscribes to those events sent by the OMS related to the creation and deletion of norms (i.e., $registerNorm$ and $deregisterNorm$ events). Thus, when the NM receives an event informing about the creation of a new norm, then it adds this norm into its norm list and subscribes to the event that activates the norm (i.e., it sends the *subscription* event to the TM with the event template A[2]).

When a norm is deregistered, then the NM removes it from its norm list. Moreover, it removes all instances that have been created out of this norm. For each deleted instance, the NM unsubscribes from its expiration event (i.e., it sends the *unsubscription* event to the TM with the event template E') and generates an event for informing about the deletion of this instance (i.e., a *normDeletion* event is sent through the event sending box).

[2]For simplicity we omit the time at which events are generated

9.4.1.2 Instance Management

Once the activation event of a norm is received (i.e., $unification(e, A)$ is true), then the NM instantiates the norm (i.e., $instantiation(e, n)$) and adds it to the instance list. At this moment, the NM subscribes to the expiration event and informs about the activation of the norm (i.e., the *instanceActivation* event is sent by the NM).

Similarly, when the NM receives the expiration event of any instance (i.e., $unification(e, E')$ is true), then it removes the instance from the instance list, unsubscribes from the expiration event and informs about the expiration of this instance (i.e., the *instanceExpiration* is sent by the NM).

Initially, there is a single NM registered in the Magentix2 platform. However, the NM is capable of simple adaptation behaviours (i.e., replication and death) in response to changing situations. For example, before the NM collapses (i.e., its event reception box is full), it might replicate itself and unsubscribe from the $registerNorm$ event. Thus, the new NM is responsible for controlling the activation of the new norms. Similarly, if the NM reaches a state in which it has no norm to control and it is not the last NM subscribed to the $registerNorm$ event, then it removes itself. These replication and death mechanisms are a simple example that illustrates how the highest layer of MaNEA can be dynamically distributed into several NMs. However, the definition of more elaborated procedures for adapting dynamically to changing environments [NS05] is a complex issue that is out the scope of this chapter.

9.4.2 Norm Enforcer

The Norm Enforcer (NE) is responsible for controlling agent behaviour. Thus, it detects violations and fulfilments of norms, and reacts upon it by sanctioning or rewarding agents. Algorithm 2 illustrates the control loop executed by the NE. As illustrated by this algorithm, the NE maintains a list (I) with the instances that hold at a given moment. Thus, it subscribes to the events sent by the NM that inform about the activation and expiration of instances, and the deletion of norms. Besides that, the NE is also in charge of controlling agents affected by the instances. Thus, it maintains a list P that contains all powers that have been created out of instances. To determine which agents are controlled by these instances, it also maintains a list (RE) containing information about role enactment (i.e., the set of roles that each agent is playing at a given moment). Thus, the NE subscribes to the events sent by the OMS that

Algorithm 1 Norm Manager Control Loop

Require: Event reception box E_{In}
Require: Event sending box E_{Out}
Require: Norm list N
Require: Instance list I
1: Add $\langle subscription, NM, \langle registerNorm, OMS, - \rangle \rangle$ to E_{Out}
//where NM stands for Norm Manager
2: Add $\langle subscription, NM, \langle deregisterNorm, OMS, - \rangle \rangle$ to E_{Out}
3: **while** E_{In} is not empty **do**
4: Retrieve e from E_{In} // $e = \langle Type, Time, Origin, Data \rangle$
//Norm Management
5: **if** $Type = registerNorm$ **then**
// $Data = \langle id, D, T, A, E, C, S, R \rangle$
6: Add $Data$ to N
7: Add $\langle subscription, NM, A \rangle$ to E_{Out}
8: **end if**
9: **if** $Type = deregisterNorm$ **and** $Data$ **in** N **then**
// $Data = \langle id, D, T, A, E, C, S, R \rangle$
10: Remove $Data$ from N
11: Add $\langle unsubscription, NM, A \rangle$ to E_{Out}
12: **for all** i **in** I **do**
// $i = \langle id', D', T', E', C', S', R' \rangle$
13: **if** $id' = id$ **then**
14: Remove i from I
15: Add $\langle unsubscription, NM, E' \rangle$ to E_{Out}
16: Add $\langle normDeletion, NM, i \rangle$ to E_{Out}
17: **end if**
18: **end for**
19: **end if**
//Instance Management
20: **for all** n **in** N **do**
// $n = \langle id, D, T, A, E, C, S, R \rangle$
21: **if** $unification(e, A) =$ **true then**
// *the norm is active*
22: $i = instantiation(e, n)$
//$i = \langle id', D', T', E', C', S', R' \rangle$ *is an instance*
23: **if** i **not in** I **then**
24: Add i to I
25: Add $\langle instanceActivation, NM, i \rangle$ to E_{Out}
26: Add $\langle subscription, NM, E' \rangle$ to E_{Out}
27: **end if**
28: **end if**
29: **end for**
30: **for all** i **in** I **do**
// $i = \langle id', D', T', E', C', S', R' \rangle$
31: **if** $unification(e, E') =$ **true then**
32: Remove i from I
33: Add $\langle unsubscription, NM, E' \rangle$ to E_{Out}
34: Add $\langle instanceExpiration, NM, i \rangle$ to E_{Out}
35: **end if**
36: **end for**
37: **end while**

inform about the fact that an agent has acquired or left a role (*acquireRole* and *leaveRole* events). In addition, the NE also subscribes to the *expel* event, which informs about the fact that a particular agent has been forced to leave a role as a disciplinary measure.

Algorithm 2 Norm Enforcer Control Loop

Require: Event reception box E_{In}
Require: Event sending box E_{Out}
Require: Instance list I
Require: Power list P
Require: Role enactment list RE
1: Add $\langle subscription, NE, \langle instanceActivation, NM, -\rangle\rangle$ to E_{Out}
//where NE stands for Norm Enforcer
2: Add $\langle subscription, NE, \langle instanceExpiration, NM, -\rangle\rangle$ to E_{Out}
3: Add $\langle subscription, NE, \langle normDeletion, NM, -\rangle\rangle$ to E_{Out}
4: Add $\langle subscription, NE, \langle acquireRole, OMS, -\rangle\rangle$ to E_{Out}
5: Add $\langle subscription, NE, \langle leaveRole, OMS, -\rangle\rangle$ to E_{Out}
6: Add $\langle subscription, NE, \langle expel, OMS, -\rangle\rangle$ to E_{Out}
7: **while** E_{In} is not empty **do**
8: Retrieve e from E_{In} // $e = \langle Type, Time, Origin, Data\rangle$
//Role enactment management
// ...(See Algorithm 3)
//Instance management
// ...(See Algorithm 4)
//Observation of Behaviour
// ...(See Algorithm 5)
75: **end while**

As in case of the NM, the NE starts retrieving an event from its event reception box. Then, different operations are performed according to the type of the event received. Specifically, the NE carries out a process that can be divided into three different activities: role enactment management, instance management and observation of behaviours.

9.4.2.1 Role Enactment Management

Algorithm 3 illustrates the pseudocode corresponding to the role enactment management process. Specifically, when the OMS informs that an agent (identified by $AgentID$) has acquired a new role (identified by $RoleID$), then the NE updates the role enactment list. Moreover, the list of instances is also checked for determining which instances affect the role $RoleID$. For each one of these instances, the NE creates a new power addressed to the agent identified by $AgentID$. In addition, the NE subscribes to the event expressed in the norm condition in order to be aware of the fulfilment or violation of this norm; i.e., it requests its subscription to the events that match the template $C'' = \langle C'_{Type}, AgentID, C'_{Data}\rangle$.

When the NE is informed by the OMS about the fact that an agent (identified by $AgentID$) is not longer playing a role (identified by $RoleID$) (i.e., *leaveRole* or *expel* events are received

by the NE), then the role enactment list is updated. Similarly, all powers that affect the agent $AgentID$ as a consequence of being playing the role $RoleID$ are removed. Therefore, the NE does not have to observe the norm condition anymore and unsubscribes from this event. Finally, if any agent leaves a role voluntarily (i.e., the *leaveRole* event is received) before fulfilling its pending obligations, then it is sanctioned (i.e., the NE performs the sanctioning action S''). The definition of actions that are applied as sanctions and rewards are domain dependent. For example, these sanctions might consist on a degradation of the public evaluation of a seller (as occurs in eBay[3]), or malicious agents may be expelled from the organization by the infrastructure itself, or there may be other domain agents in charge of performing sanctions. Besides that, the NE informs about the fact that an agent has been sanctioned for violating an obligation (i.e., the *sanction* event is sent through the E_{Out} box).

Algorithm 3 Role Enactment Management

```
 9: if Type = acquireRole then
    // Data is a pair (AgentID, RoleID)
10:   Add Data to RE
11:   for all i in I do
       // i = ⟨id', D', T', E', ⟨C'_Type, −, C'_Data⟩, S', R'⟩
12:     if T' = RoleID then
13:       C'' = ⟨C'_Type, AgentID, C'_Data⟩
14:       Add ⟨id', D', T', C'', S', R', false⟩ to P
15:       Add ⟨subscription, NE, C''⟩ to E_Out
16:     end if
17:   end for
18: end if
19: if Type = leaveRole or Type = expel then
    // Data is a pair (AgentID, RoleID)
20:   Remove Data from RE
21:   for all p in P do
       // p = ⟨id'', D'', T'', C'', S'', R'', W''⟩ and C'' = ⟨C''_Type, AgentID'', C''_Data⟩
22:     if T'' = RoleID and AgentID'' = AgentID then
23:       Remove p from P
24:       Add ⟨unsubscription, NE, C''⟩ to E_Out
25:       if D'' = O and W'' = False and Type = leaveRole then
           // O stands for obligation
26:         Perform S''// against AgentID
27:         Add ⟨sanction, NE, violated(id'', AgentID)⟩ to E_Out
28:       end if
29:     end if
30:   end for
31: end if
```

[3]http://www.ebay.com/

9.4.2.2 Instance Management

This process is contained in Algorithm 4. When the NE is informed by the NM about the creation of a new instance (i.e., the NE receives the *instanceActivation* event), then the NE updates its instance list and creates new powers for controlling all the agents that are playing the target role at that moment. The watch condition (W'') of powers is initially set to false. Thus, for each one of the new powers the NE starts to observe indirectly norm compliance by subscribing to the event C''.

Algorithm 4 Instance Management

32: **if** $Type = instanceActivation$ **then**
    // $Data = \langle id', D', T', E', \langle C'_{Type}, -, C'_{Data}\rangle, S', R'\rangle$
33: Add $Data$ to I
34: **for all** $(AgentID, RoleID)$ **in** RE **do**
35: **if** $RoleID = T'$ **then**
36: $C'' = \langle C'_{Type}, AgentID, C'_{Data}\rangle$
37: Add $\langle id, D', T', C'', S', R', \textbf{false}\rangle$ to P
38: Add $\langle subscription, NE, C''\rangle$ to E_{Out}
39: **end if**
40: **end for**
41: **end if**
42: **if** ($Type = instanceExpiration$ **or** $Type = normDeletion$) **and** $Data$ *in* I **then**
    // $Data = \langle id, D, T, E, C, S, R\rangle$
43: Delete $Data$ from I
44: **for all** p **in** P **do**
    // $p = \langle id'', D'', T'', C'', S'', R'', W''\rangle$ *and* $C'' = \langle C''_{Type}, AgentID, C''_{Data}\rangle$
45: **if** $id' = id''$ **then**
46: Remove p from P
47: Add $\langle unsubscription, NE, C''\rangle$ to E_{Out}
48: **if** $Type = instanceExpiration$ **then**
    // *The agent is responsible for norm fulfilment*
49: **if** $W'' =$ **false and** $D'' = \mathcal{O}$ **then**
    // *The obligation has not been fulfilled before it has expired*
50: Perform S'' // *against* $AgentID$
51: Add $\langle sanction, NE, violated(id'', AgentID)\rangle$ to E_{Out}
52: **end if**
53: **if** $W'' =$ **false and** $D'' = \mathcal{F}$ **then**
    // *The prohibition has been observed*
54: Perform R'' // *in favour of* $AgentID$
55: Add $\langle reward, NE, fulfilled(id'', AgentID)\rangle$ to E_{Out}
56: **end if**
57: **end if**
58: **end if**
59: **end for**
60: **end if**

When an instance has no longer effect (i.e., the NE receives the *instanceExpiration* or *normDeletion* event), then the NE updates the instance list and removes all powers created out of this instance. An instance becomes ineffective whenever its expiration condition holds or the norm that has given rise to it is abolished. In the first case (i.e., the NM receives the

instanceExpiration event), the agents controlled by this instance are responsible for fulfilling the norm. Thus, if the instance obliges agents to reach some state of affairs (e.g., agents are obliged to perform an action) and this state has not been observed yet (i.e., the watch condition W'' of powers is false), then the offender agents are sanctioned. On the contrary, if agents are prohibited to reach some situation and the forbidden state has not been observed (i.e., W'' of powers is false), then agents are rewarded. Finally, if an instance becomes ineffective due to the deletion of a norm, then agents are not responsible for the fulfilment of the norm and enforcement actions are not performed.

Algorithm 5 Observation of Behaviours

```
61: for all p in P do
    // p = ⟨id'', D'', T'', C'', S'', R'', W''⟩ and C'' = ⟨C''_Type, AgentID, C''_Data⟩
62:   if unification(e, C'') = true then
63:     Remove p from P
64:     if D'' = F then
          // The prohibition has been violated
65:       Add ⟨id'', D'', T'', C'', S'', R'', true⟩ to P
66:       Perform S'' // against AgentID
67:       Add ⟨sanction, NE, violated(id'', AgentID)⟩ to E_Out
68:     else
          // The obligation has been fulfilled and it expires
69:       Perform R'' // in favour of AgentID
70:       Add ⟨reward, NE, fulfilled(id'', AgentID)⟩ to E_Out
71:       Add ⟨unsubscription, NE, C''⟩ to E_Out
72:     end if
73:   end if
74: end for
```

9.4.2.3 Observation of Behaviours

This functionality is implemented by Algorithm 5. The NE checks for each one of the powers whether the C'' event has been detected (i.e., $unification(e, C'')$ is true). If it is the case, then the power is updated. The watch condition is registered as true indicating that the norm condition has been perceived. Then, enforcement actions are performed according to the deontic modality of the power. For example, if the power is an obligation, then the obligation is considered as fulfilled (i.e., the power is deleted from P) and the agent is rewarded. Similarly, if it is a prohibition, then the agent is sanctioned. However, in case of a prohibition the power is not removed until the norm expires, since sanctions must be applied if the forbidden event is detected again.

As in case of the NM, the lowest level of MaNEA has been described assuming that there is

a single NE. However, this layer may be formed by a set of specialized NEs. For example, the set of instances can be distributed among NEs according to the target role. Thus, each NE is responsible for controlling actions in which a specific set of roles is involved. It is also possible to specialize NEs for controlling a specific group of agents independently of the roles that they play. Finally, it is also possible to dynamically adapt the amount of NEs by performing cloning and self-deletion operations.

9.5 Implementation of the n-BDI Architecture

In this section we describe how a prototype of the n-BDI architecture has been developed using Jason and Magentix2. This section is structured as follows: Section 9.5.1 contains an overview of the Jason interpreter. In Section 9.5.2 the prototype of the n-BDI architecture implemented in Jason [BHW08] is briefly described.

9.5.1 Jason

Jason [BHW08] is an interpreter for an extended version of the agent-oriented language AgentSpeak [Rao96] that gives support to the creation of BDI agents.

Jason agents [BHW08] operate by means of a *reasoning cycle* which can be divided into 10 main steps (see Figure 9.1):

Step 1. Perceiving the Environment. The first thing an agent does within a reasoning cycle is to sense the environment so as to update its beliefs about the state of the environment. The `perceive` method is used to implement the process of obtaining such percepts.

Step 2. Updating the Belief Base. Once the list of percepts has been obtained, the belief base needs to be updated to reflect perceived changes to the environment. This is done by the `buf` method.

Step 3. Receiving Communication. At this stage, the interpreter checks for messages that might have been delivered to the agent's "mailbox". This is done by the `checkMail` method.

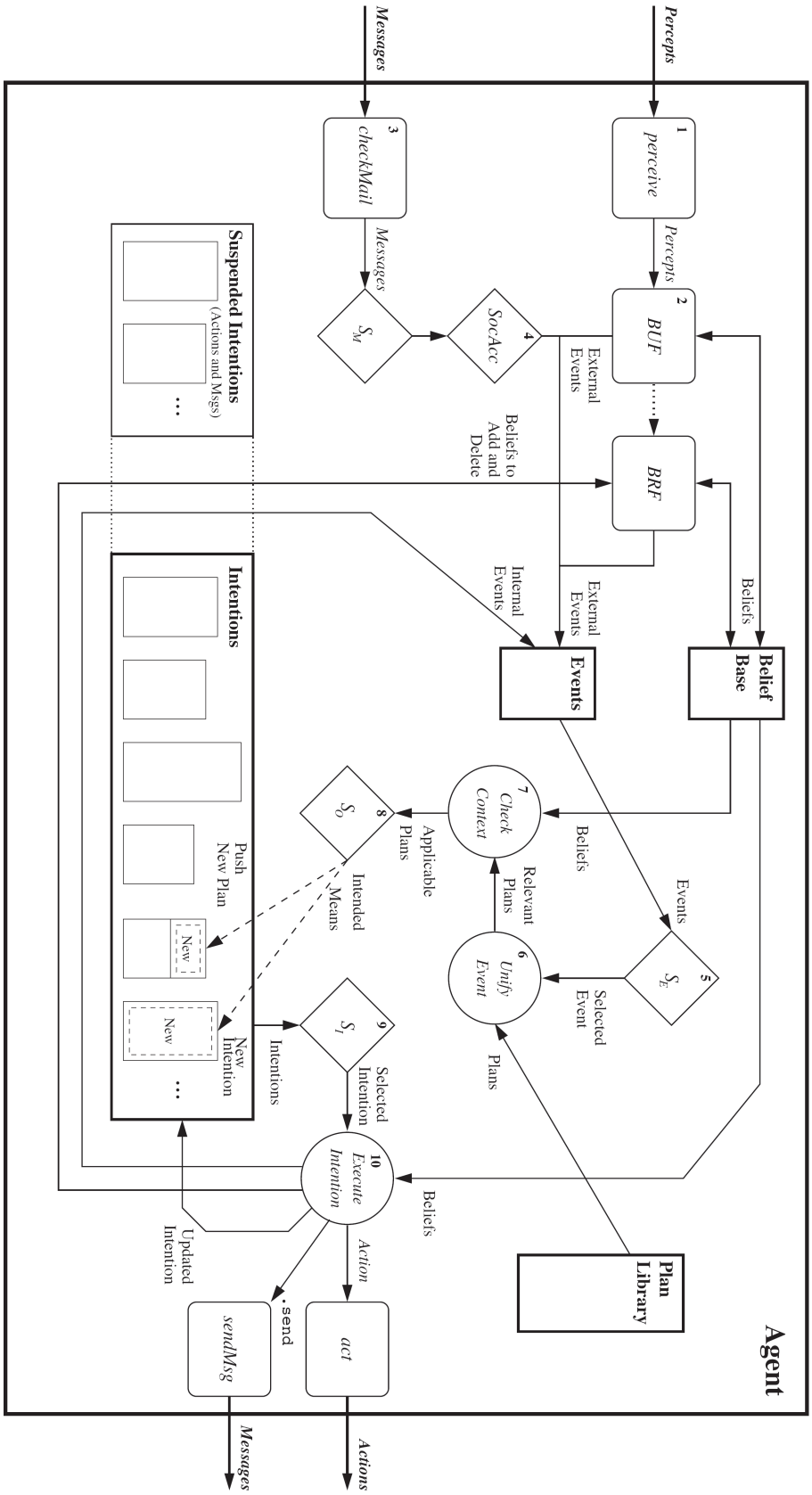

Figure 9.1: The Jason reasoning cycle [BHW08]

Step 4. Selecting "Socially" Acceptable Messages. Before messages are processed, they go through a selection process to determine whether they can be accepted by the agent or not. A method named `SoccAcc` is responsible for this process.

Step 5. Selecting an Event. Practical BDI agents operate by handling events, which represent either perceived changes in the environment or in the agent's goals. Therefore, Jason agents need to select an event to be handled in a particular reasoning cycle. This is done by the event selection function S_ε. A method named `selectEvent` is responsible for this process.

Step 6. Retrieving all Relevant Plans. Once the agent has selected an event, it needs to find a plan that allows the agent to act so as to handle that event.

Step 7. Determining the Applicable Plans. At this stage the agent determines which relevant plans can be used at this moment given the information that it currently has.

Step 8. Selecting One Applicable Plan. The selection of a particular plan, from the set of applicable plans, that will be included into the set of intentions is done by the option selection function (named S_O). A method named `selectOption` is responsible for this process.

Step 9. Selecting an Intention for Further Execution. At this stage, the agent selects the next intention to be executed. A method named `selectIntention` is responsible for this process.

Step 10. Executing One Step of an Intention. Finally, the agent acts upon the environment.

For a complete description of the Jason reasoning cycle see [BHW08].

9.5.1.1 Jason in Magentix2

Magentix2 provides native support for executing Jason agents anf this framework has been integrated into Magentix2. Therefore, we can program agents in AgentSpeak and run them on Magentix2 platform. Thereby, Jason agents can benefit from the reliable communication,

tracing facilities and security mechanisms provided by Magentix2. For these reasons we have selected the Jason interpreter as a basis for implementing a prototype of n-BDI agent architecture in Magentix2.

Magentix2 integrates Jason providing two classes: `MagentixAgArch` and `JasonAgent`. The `MagentixAgArch` class manages the AgentSpeak interpreter, the reasoning cycle of the agent, and how the agent acts and perceives to/from the environment. The `JasonAgent` class acts as a link between the AgentSpeak interpreter and the platform. Both classes can be modified and adapted to the desired needs.

In this chapter we propose the extension of the `MagentixAgArch` class and the creation of a new agent class (`MagentixNBDIAgent`) to allow agents to have an explicit representation of deontic norms and instances and to consider them in their decisions according to the n-BDI architecture.

9.5.1.2 Implementations of Normative Agents in Jason

The implementation of normative BDI agents is not new. In fact, there is a previous work [dSNdSdL11] in which Jason has been used for developing normative agents. This work has been considered as a reference for the implementation described in this chapter. The main difference among the two implementations is the fact that n-BDI agents are capable of considering norms within uncertain and dynamic environments. This implies that agents have not a perfect knowledge of the world and that the set of norms that regulate the agents' environment may change along time.

9.5.2 Implementing the n-BDI Architecture in Magentix2 using Jason

The implementation of the n-BDI architecture in Jason has been carried out by modifying the functions that perform the steps of the norm reasoning cycle (see Section 4.2.4). The changes that have been made to these functions are described below.

9.5.2.1 Receiving Communication: Acquiring Norms

As previously mentioned, `checkMail` is the method that makes available the messages received by the underling infrastructure and at the level of the Jason interpreter.

In the n-BDI model the messages that are sent by the OMS and that inform about the modification of the normative system (i.e., the register and deregister of norms) are considered for updating the set of norms and instances that are managed by the agent. This functionality is implemented by Function 9.1. Thus, when the agent receives a message in which it is informed by the OMS about the register of a new norm, then the agent updates the list of abstract norms by adding the new norm. Similarly, when the agent receives a message in which it is informed about the deregister of an existing norm, then the agent updates the set of abstract norms by deleting this norm. Moreover, it deletes all instances that have been created out of the norm that has been reregistered. This function has been included in the `MagentixAgArch` class.

In this implementation we assume that only the OMS is allowed to inform about the norms that regulate a VO. Moreover, norms are equally salient. If this is not the case, then the agent must store the norm opinions and calculate the salience of norms as explained in Section 4.3.2.

9.5.2.2 Belief Revision: Norm Relevance

The method `buf` updates the belief base with the given percepts and adds all changes that occurred as new events in the set of events.

In the n-BDI implementation, we have overridden this method (see Function 9.2) for determining which norms are relevant to the agent (`selectRelevantNorms(roles)`), which of the relevant norms have been instantiated (`selectActivateNorms()`), and which instances expired according to the agent beliefs (`this.selectExpiratedNorms()`). For example, Function 9.3 determines which of the relevant norms have been instantiated and updates the list of instances that are considered by the agent. This function has been added in the new agent class `MagentixNBDIAgent`.

Jason allows the annotation of beliefs to represent the certainty of these beliefs. In the current implementation agents do not consider the certainty of beliefs and all instances are equally relevant. As future work we plan to use this annotations to calculate the relevance degree of instances as defined in Section 4.4.2.1.

9.5.2.3 Selecting an Event: Reasoning About Deontic Norms

The `selectEvent` method selects the event that is handled in the current reasoning cycle. The default implementation removes and returns the first event in the queue.

In the n-BDI implementation the `selectEvent` function has been overridden to deter-

Java Function 9.1: checkMail Function

```
public void checkMail() {
  ACLMessage m;
  do {
    m = messageList.poll();
    ...
    Object propCont = translateContentToJason(m);
    ...
    //Norm Acquisition
    Literal l = (Literal) propCont;
    // If it is a norm.
    if (l.getFunctor().equals("normspecification")) {
        if (l.getTerm(6).toString().equals("registerNorm")) {
            //A new norm has been registered
            // Adds to list of abstract norms
            this.ag.addAbstractNorm(l);
        }
        if (l.getTerm(6).toString().equals("deregisterNorm")){
          //A Norm has been deregistered
          ArrayList<Literal> lListAux = (ArrayList<Literal>)this.ag.
              getinstantiatedNorms().clone();
          //Literal lToRemove = null;
          for(Literal laux : lListAux)
          {
            if (laux.getTerm(4).toString().equals(l.getTerm(5).toString()))
            {
              this.ag.removeinstantiatedNorm(laux);
              //An instance is removed
            }
          }
          ArrayList<Literal> lListAuxAbstract = (ArrayList<Literal>)this.ag.
              getAbstractNorms().clone();
          for (Literal laux : lListAuxAbstract)
          {
            if (laux.getTerm(5).toString().equals(l.getTerm(5).toString()))
            {
               this.ag.removeAbstractNorm(laux);
            }
          }
       }
    }
  }
  while (m != null);
}
```

mine which norms will be complied with and to internalize norms. The source code of this method is contained in Function 9.4. This function has been included in the new agent class `MagentixNBDIAgent`.

Norm-based Expansion. This process is performed by two functions: `calculateWillingness` and `annotatePlans`. The fist one is responsible for determining the agent willingness to com-

Java Function 9.2: Belief Update Function

```
public void buf(List<Literal> arg0) {
  if (arg0 != null) {
    super.buf(arg0);
    List<Literal> roles = getRoles();
    // Extract abstract norms with the same role
    this.selectRelevantNorms(roles);
    this.selectActivateNorms();
    this.selectExpiratedNorms();
  }
}
```

Java Function 9.3: Select Activated Norms Function

```
private void selectActivateNorms(List<Literal> percepts) {
  for (Literal literal : percepts) {
    literal.addAnnot(liaux);
    this.getBB().add(literal);
  }
  String activation = "";
  String condition = "";
  String expiration = "";
  String sanction = "";
  String reward = "";
  for (Literal rNorm : this.relevantNorms) {
    LogicalFormula lfActivation = (LogicalFormula) rNorm.getTerm(3);
    Iterator<Unifier> uActivation = lfActivation.logicalConsequence(
        this, new Unifier());
    if (uActivation.hasNext()) {
      while (uActivation.hasNext()) {
        Term term = uActivation.next().getAsTerm();
        condition = this.replaceValue(rNorm.getTerm(1), term);
        activation = this.replaceValue(rNorm.getTerm(3), term);
        expiration = this.replaceValue(rNorm.getTerm(4), term);
        sanction = this.replaceValue(rNorm.getTerm(5), term);
        reward = this.replaceValue(rNorm.getTerm(6), term);
        Literal instantiatedNorm = Literal
            .parseLiteral("instantiatednorms("
                + rNorm.getTerm(0) + ", "
                + condition + ", "
                + rNorm.getTerm(2) + ", "
                + activation + ", "
                + expiration + ", "
                + sanction + ", "
                + reward + ").");
        this.addNewInstantiatedNorm(instantiatedNorm);
      }
    }
  }
}
```

Java Function 9.4: Select Event Function

```
public Event selectEvent(Queue<Event> events){
  //Building the goal set
  List<Plan> planLibrary=super.getPL().getPlans();
  List<Term> goals=new ArrayList<Term>();
  for(Plan plan:planLibrary){
      goals.add(plan.getTrigger().getTerm(1));
  }
  //Norm-based expansion
  this.compliedNorms=calculateWillingness(this.instantiatedNorms,goals);
  annotatePlans(compliedNorms,planLibrary);
  return super.selectEvent(events);
}
```

ply with instances. The second one is responsible for annotating those plans according to the complied instances.

Function 9.5 contains the source code of the `calculateWillingness` method. For simplicity, this function calculates the willingness to comply with norms by considering only self-interest motivations and the expectations of being rewarded or sanctioned. As future work, we will extend this function to consider the emotional factor as defined in Section 5.3.

The `annotatePlans` method annotates plans according to the set of complied norms. Thus, the priority of the plans that achieve a state that is obliged by a norm is increased. On the contrary, the priority of the plans that achieve a forbidden state is decreased. As future work, we plan to extend this method to annotate plans by considering the salience and relevance of norms when plans are annotated as defined in Section 5.2.

9.5.2.4 Selecting One Applicable Plan: Action Selection

The `selectOption` function is used to select one among several options (an option is an applicable plan and an unification) to handle an event. In the n-BDI proposal it has been overridden to select the plan with the highest priority.

9.6 Case Study

To illustrate the performance of MaNEA, an example of the assembly line case-study, which has been introduced in Section 9.3.2.1, is explained below. Let us assume the existence of four domain agents: the supervisor (s_1), the auditor (a_1) and two robot agents (r_1 and r_2) that play the subordinated role. Moreover, there are infrastructural agents: the norm manager (NM),

Java Function 9.5: Calculate Willingness Function

```
private ArrayList<Literal> calculateWillingness(List<Literal> norms,List<
    Term> goals){
  ArrayList<Literal>compliedNorms=new ArrayList<Literal>();
  Unifier unifier= new Unifier();
  for(Literal norm : norms){
    int comply=0;
    //Self-interest
    if(norm.getTerm(0).toString().trim().equalsIgnoreCase("obligation")){
      for(Term goal:goals){
        if(unifier.unifies(norm.getTerm(2),goal))
                comply++;
      }
    }
    else{
      boolean desiredState=false;
      for(Term goal:goals){
          if(unifier.unifies(norm.getTerm(2),goal))
                  desiredState=true;
      }
      if(!desiredState) comply++;
    }
    //Expectations
    for(Term goal:goals){
      if(unifier.unifies(norm.getTerm(5),goal)) comply--;
      if(unifier.unifies(norm.getTerm(6),goal)) comply++;
    }
    if(comply>=0) compliedNorms.add(norm);
  }
  return compliedNorms;
}
```

the norm enforcer (NE), the trace manager (TM) and the organization management system (OMS).

Figure 9.2 shows the exchange of events among the agents TM, OMS, NM, NE and s_1 corresponding to the activation of norm n_4, which has been defined in Section 9.3.2.1 as follows:

$$\langle n_4, \mathcal{O}, subordinated,$$
$$\langle auditStart, -, task(T)\rangle, \langle auditEnd, -, task(T)\rangle,$$
$$\langle taskPerformance, -, performance(P) \wedge task(T)\rangle,$$
$$-, -\rangle$$

Specifically, agent OMS sends an event for informing the NM about the register of norm n_4. According to Algorithm 1 (described in Section 9.4.1), the NM sends an event to the TM for subscribing to the norm activation event ($\langle auditStart, -, task(T)\rangle$). The supervisor agent (s_1) is responsible for initiating an audit stage. Specifically, agent s_1 sends an event for starting the

audit of the assembling task. Then the NM receives this event and informs the NE about the creation of an instance named $i_{assembling}$:

$$i_{assembling} = \langle n_4, \mathcal{O}, subordinated,$$
$$\langle auditStart, -, task(assembling)\rangle, \langle auditEnd, -, task(assembling)\rangle,$$
$$\langle taskPerformance, -,$$
$$performance(P) \wedge task(assembling)\rangle, -, -\rangle$$

Moreover, the NM requests to the TM its subscription to the expiration event ($\langle auditEnd, -, task(assembling)\rangle$). Since there are two agents (r_1 and r_2) that are playing the subordinated role, two powers are created. Thus, the NE executes Algorithm 4 (described in Section 9.4.2.2) and sends two subscription requests to the TM for controlling these two powers.

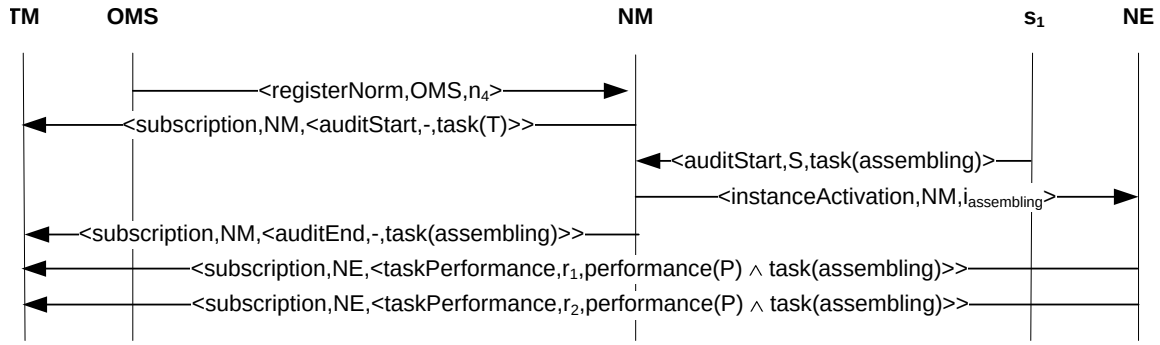

Figure 9.2: Norm activation

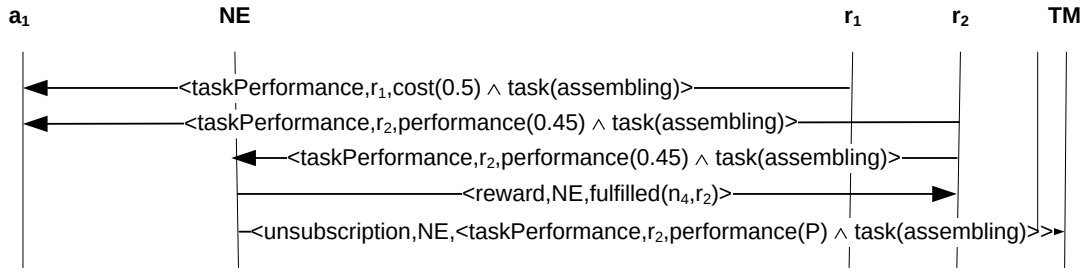

Figure 9.3: Observation of behaviours

Figure 9.3 corresponds to the exchange of events that occur when the norm n_4 is active. When the audit stage of the assembling task starts, agents r_1 and r_2 send events for informing about this task. Since the auditor agent (a_1) is subscribed to this information, it receives these events. On the one hand, agent r_1 informs about the cost of this task. This event does not

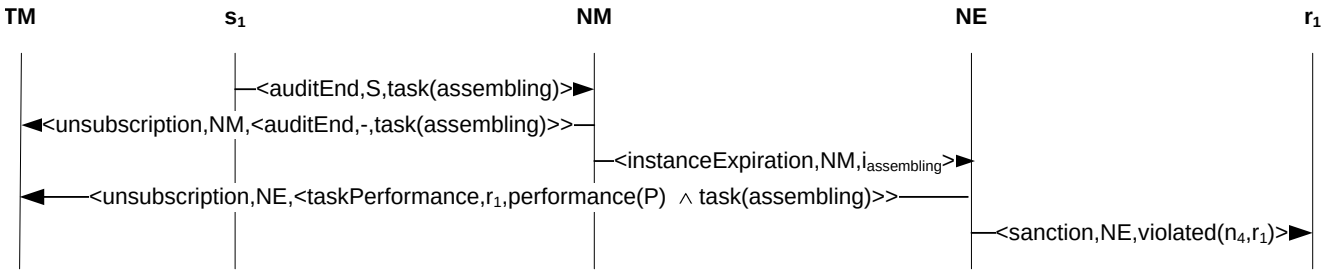

Figure 9.4: Norm expiration

match the norm condition template and, as a consequence, agent NE does not receive this event. On the other hand, agent r_2 informs about the performance of this task. In this case, agent NE receives this event and it considers that agent r_2 has fulfilled the norm n_4. Agent NE executes Algorithm 5 (described in Section 9.4.2.3), rewards agent r_2 and unsubscribes from the event that controls the power corresponding to r_2.

Finally, Figure 9.4 illustrates the events that are sent when the instance $i_{assembling}$ expires. This process starts when the agent s_1 sends the expiration event ($\langle auditEnd, -, task(assembling)\rangle$). Then the NM unsubscribes from this event, deletes the instance and informs the NE about the expiration of the instance $i_{assembling}$. The NE executes Algorithm 4 (described in Section 9.4.2.2) and checks all powers related to the instance that remain unfulfilled. Specifically, the NE removes the power that controls agent r_1 and unsubscribes itself from the norm condition event ($\langle taskPerformance, r_1, performance(P) \wedge task(assembling)\rangle$) and sends a sanctioning event to agent r_1, since this agent has not fulfilled the obligation.

As illustrated by this example, the supervisor agent is responsible for starting and ending the audit stages. Similarly, the auditor agent gathers the information provided by subordinated agents to analyse the assembly line. Thus, neither the supervisor nor the auditor must control norms. The designer of this case study does not have to program any agent that is responsible for controlling norms since the infrastructure itself provides this functionality. Agents r_1 and r_2 must take into account norms if they want to avoid sanctions. Thus, the agent designer must program norms on agents [KN03] or must endow these agents with capabilities for accepting these norms while maintaining their autonomy [CAB10a]. Specifically, the previous section describes how a prototype of the n-BDI architecture has been implemented in Magentix2.

9.7 Evaluation

One of the main novelties of MaNEA is that it is based on a tracing service, which has been implemented following a publish/subscribe metaphor. Traditionally, Norm-Enforcing Architectures have been built using overhearing approaches. Overhearing is defined as an indirect interaction whereby an agent receives information for which it is not addressee [KPT02, LT03]. In this section the MaNEA proposal is evaluated theoretically and experimentally in order to illustrate its performance with respect to these overhearing approaches. Specifically, MaNEA is compared with two other proposals: Cardoso & Oliveira' approach, and Modgil et al. framework [MFM+09].

Cardoso & Oliveira' approach is a centralized overhearing approach (or centralized approach for short) in which exists a centralized entity (known as *norm environment*) that receives information about all the messages that agents exchange. This entity considers these messages together with the norms to determine if any agent has violated or fulfilled a norm. This proposal has been selected to evaluate the performance of MaNEA with respect to the number of messages exchanged due to norm enforcement since it is a centralized approach. Centralized approaches are supposed to require the optimum number of messages.

Modgil et al. proposal is a mixed approach that uses overhearing and subscription approaches. In particular, the monitor subscribes to observers that report states that are relevant to norm reasoning (i.e. states that are included in the norms). All messages and actions performed by agents are reported to a set of trusted observers. As Table 9.1 (in Section 9.2.2) shows, Modgil et al. framework is the proposal that provides a functionality similar to MaNEA; i.e., it also enforces general norms that control actions, messages and states of affairs that may occur independently. Moreover, it also proposes a two layer architecture for enforcing norms. Therefore, this proposal has been selected among the distributed norm-enforcing architectures to evaluate the performance of MaNEA with respect to the number of messages exchanged due to norm enforcement.

9.7.1 Theoretical Results

Let us consider a general scenario in which we have a set of agents ($\mathcal{A}$) that interact over the course of I iterations. In each iteration each agent performs one action. Each agent plays one or more roles. For simplicity we make the following assumptions: (i) we do not take into

account the fact that agents may play different roles during their execution; (ii) each agent is randomly assigned to a set of roles in each execution; (iii) this scenario is controlled by a set of norms; and (iv) norms are not changed at runtime; i.e., we assume that there is a set of norms ($\mathcal{N} = \{n_1, ..., n_j\}$) that remains static.

9.7.1.1 Cardoso & Oliveira

In Cardoso & Oliveira' approach [CO07] agents report all messages that they sent to a single entity, which will be named agent C, that is responsible for norm enforcement. For simplicity, we assume that agents also report all actions that they perform to C. This entity processes the reporting messages together with the norms to infer norm violations and fulfilments.

To determine the sequence and number of messages that are exchanged in this approach, let us start with a simple example depicted in Figure 9.5. This figure illustrates the message exchange among a set of agents ($\{C, A1, A2, A3, A4\}$) along the different stages of norm monitoring (*Initialization*, *Activation*, $Fulfilment$ and *Expiration*). Agents $A1, A2, A3$ and $A4$ are domain agents. We consider the case in which C is only responsible for monitoring an obligation norm ($\langle id, \mathcal{O}, T, A, E, C, -, - \rangle$). When agent $A2$ performs the action that activates the norm (Figure 9.5 message $action(A)$) then the norm is instantiated ($\langle id', \mathcal{O}, T', E', C', -, - \rangle$). From that moment on, agent C controls all of the agents that are playing role T'. Let us assume that agents $A3$ and $A4$ are playing that role so that they are under the influence of this norm and two powers are created for controlling these agents ($\langle id'', \mathcal{O}, T'', E'', C'', -, -, W'' \rangle$). When agent $A3$ performs the obliged action, it sends a reposting message to agent C (Figure 9.5 message $action(C'')$). Then the norm has been fulfilled and C rewards agent $A3$. Finally, when agent $A1$ performs the expiration action (Figure 9.5 messages $action(E')$), then C sanctions $A4$ (Figure 9.5 message *sanction*), since this agent has not complied with the norm. At any step agent C receives messages that are not related to norms. These other messages report all the actions that have been performed by agents (e.g., the message reporting action X_1 in Figure 9.5).

As illustrated by this example, the number of messages that are exchanged for controlling norms in Cardoso & Oliveira' proposal is:

- *Initialization.* Agent C receives messages informing about all the actions that have been performed. Thus, c does not have to subscribe to the activation event of norms. Therefore, no message is sent for this purpose in the initialization step.

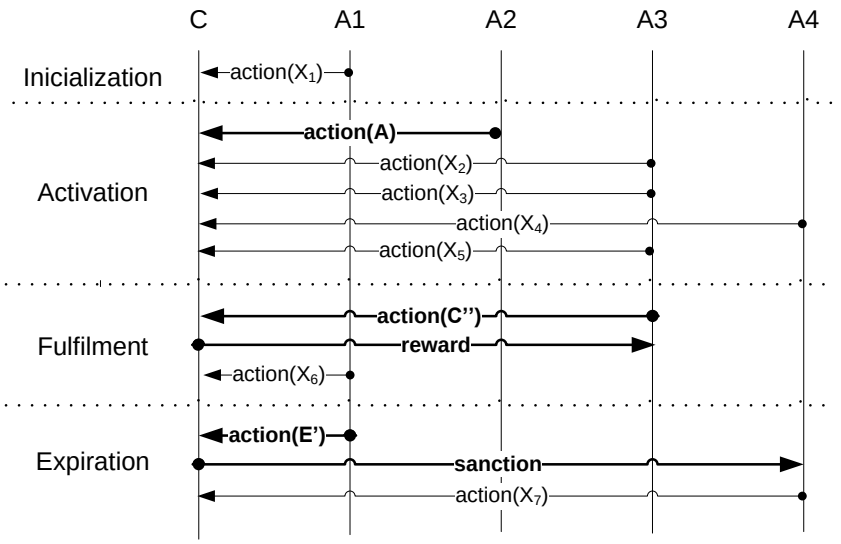

Figure 9.5: Messages exchanged in Cardoso & Oliveira' approach when a single norm is controlled

- *Activation.* When norms are activated it is not necessary to send any extra message.
- *Expiration.* Similarly to the activation stage, no extra message is sent when norms expire.
- *Fulfilment.* If an agent fulfils a norm, then agent C sends 1 rewarding message.
- *Violation.* When an agent violates a norm, then C sends 1 message for sanctioning the offender agent.
- *Reporting.* All actions that are performed by agents are reported to agent C. Therefore, agent C receives one reporting message for each action that each agent performed.

Thus, the number of messages required for controlling a single norm n when a set of agents ($\mathcal{A}$) interact along I iterations is:

$$\underbrace{0}_{Initialization} + \underbrace{0}_{Activation} + \underbrace{0}_{Expiration} + \underbrace{\theta_F^n}_{Fulfilment} + \underbrace{\theta_V^n}_{Violation} + \underbrace{|\mathcal{A}| * I}_{Reporting}$$

where θ_V^n and θ_F^n are the number of times that a norm has been violated or fulfilled, respectively. The total number of messages that are required for controlling a set of norms $\mathcal{N}(\mathcal{N} = \{n_1, ..., n_j\})$ is:

$$(|\mathcal{A}| * I) + \sum_{i=1}^{j} (\theta_F^{n_i} + \theta_V^{n_i})$$

9.7.1.2 Modgil

In the case of the proposal of Modgil et al., agents report to observers all the actions that they perform, as in case of the centralized approach. However, observers only report information to monitors when the information is relevant to the activation, expiration, violation or fulfilment of norms. Therefore, the number of messages that is required for controlling norms depends on the number of times that observers report information to monitors. To illustrate the messages that are necessary for controlling norms, we start with a simple example that is shown in Figure 9.6. This figure illustrates the message exchange among a set of agents ($\{M, O, A1, A2, A3, A4\}$) along the different stages of norm monitoring. M and O are the monitor and observer agents, respectively. Agents $A1, A2, A3$ and $A4$ are domain agents. Again, we will consider the case in which M is responsible for monitoring the same obligation norm ($\langle id, \mathcal{O}, T, A, E, C, -, -\rangle$). M subscribes to the observer entrusted with reporting on the states of interest identified by the norm (A, E, C). When agent $A2$ reports to the observer that it has performed the action that activates the norm, it sends the event that indicates that the norm is active (Figure 9.6 message $action(A)$). Then O sends this information to M and the norm is instantiated ($\langle id', \mathcal{O}, T', E', C', -, -\rangle$). From that moment on, M controls all agents that are under the influence of the norm. Again, agents $A3$ and $A4$ are affected by the norm; i.,e., two powers are created ($\langle id'', \mathcal{O}, T'', E'', C'', -, -, W''\rangle$) . Agent $A4$ performs action E'''. E''' is an instantiation of E (i.e., exits a substitution σ such as $\sigma(E) = E'''$) and O informs agent M about it. However, E''' does not match the expiration condition of the instance (i.e., $E''' \neq E'$) and the instance does not expire. When agent $A3$ performs the obliged action, then it sends a reporting message to O (Figure 9.6 message $action(C'')$). The observer sends this information to the monitor M. As previously mentioned, in case of norm violations and fulfilment the monitor takes remedial actions accordingly. In this experiment we have assumed that these remedial actions consist in sending sanctioning and rewarding actions. Thus, M rewards agent $A3$. Finally, when agent $A1$ performs the expiration action (Figure 9.6 message $action(E')$), then the observer sends this information to M. M is aware of the expiration of the obligation instance and sanctions $A4$ (Figure 9.6 message *sanction*).

As illustrated by this example, the number of messages that are exchanged in Modgil et al. framework is:

- *Initialization.* Agent M must subscribe to the observers that inform about the states of

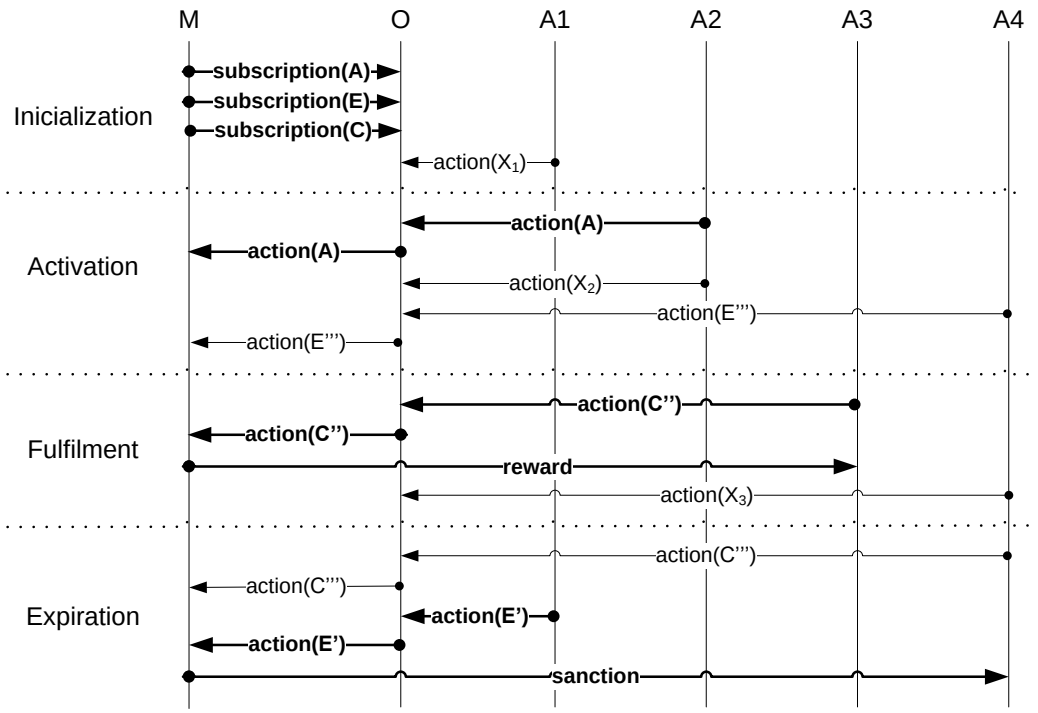

Figure 9.6: Messages exchanged in Modgil et al. approach when a single norm is controlled

interest specified by the norm. Therefore, 3 messages are sent in the initialization step.

- *Activation.* When a norm is activated 1 message is sent by an observer to the monitor.

- *Expiration.* Every time that an observer detects that the expiration condition (E) of a norm has been instantiated it sends 1 message to the monitor. If the instantiation is equal to the expiration condition of an instance (E') then this instance expires. If the instantiation is not equal to the expiration condition of any instance then all instances remain active.

- *Fulfilment.* Every time that an observer detects that an agent has performed an action that is an instantiation of the norm condition (C), then it sends 1 message to the monitor. If the instantiation is equal to the norm condition of an instance (C') and the agent that has performed the action is under the influence of the instance (i.e., it plays the target role) then the instance has been fulfilled. In this case, agent M sends 1 rewarding message. On the contrary, if the instantiation is different from C'' or the agent is not under the influence of the instance then the instance remains unfulfilled.

- *Violation.* When an agent violates a norm, then 1 message is sent for sanctioning the offender agent.

- *Reporting.* All actions that are performed by agents are reported to observers. Therefore, the observer agent receives one reporting message for each action that has been performed.

Thus, the number of messages required for controlling a single norm n when a set of agents ($\mathcal{A}$) interact along I iterations is:

$$\underbrace{3}_{Initialization} + \underbrace{\theta_A^n}_{Activation} + \underbrace{\theta_E^n}_{Expiration} + \underbrace{\theta_C^n + \theta_F^n}_{Fulfilment} + \underbrace{\theta_V^n}_{Violation} + \underbrace{|\mathcal{A}| * I}_{Reporting}$$

where θ_A^n, θ_E^n and θ_C^n are the number of times that an observer detects that an instantiation of the activation, the expiration, or the norm condition of a given norm holds, respectively. θ_V^n and θ_F^n are the number of times that a norm has been violated or fulfilled, respectively. The total number of messages that are required for controlling a set of norms ($\mathcal{N} = \{n_1, ..., n_j\}$) is:

$$(|\mathcal{A}| * I) + \sum_{i=1}^{j}(3 + \theta_A^{n_i} + \theta_E^{n_i} + \theta_C^{n_i} + \theta_F^{n_i} + \theta_V^{n_i})$$

9.7.1.3 MaNEA

In MaNEA, the number of messages exchanged depends on the number of actions that are relevant to the activation, expiration and fulfilment of norms. To illustrate the exchange of messages that occurs in MaNEA we also use a simple example that is shown in Figure 9.7. This figure illustrates the message exchange among a set of agents {NM, NE, TM, $A1, A2, A3, A4$} along the different stages of norm monitoring. NM, NE and TM are the Norm Manager, Norm Enforcer and Trace Manager. Agents $A1, A2, A3$ and $A4$ are domain agents. Again, we consider the case in which MaNEA is only responsible for monitoring the same obligation norm ($\langle id, \mathcal{O}, T, A, E, C, -, -\rangle$). Thus, the NM subscribes to the activation event of this norm (Figure 9.7 message *subscription*(A)). When agent $A2$ sends the event that indicates that the norm is active (Figure 9.7 message *event*(A)) then the norm is instantiated ($\langle id', \mathcal{O}, T', E', C', -, -\rangle$) and the NM subscribes to the expiration event (Figure 9.7 message *subscription*(E')). Moreover, the NM sends a message to the NE for informing about the creation of a new instance (Figure 9.7 message *instanceActivation*). Again, agents $A3$ and $A4$ are under the influence of the norm and two powers are created ($\langle id'', \mathcal{O}, T'', E'', C'', -, -, W''\rangle$). Thus, the NE sends two messages to the TM for subscribing to the events sent by $A3$ and $A4$ that inform about the fulfilment of the norm (Figure 9.7 messages *subscription*(C'')). When agent $A3$ performs the obliged action, then it sends a message to the NE (Figure 9.7 message *event*(C'')). Then

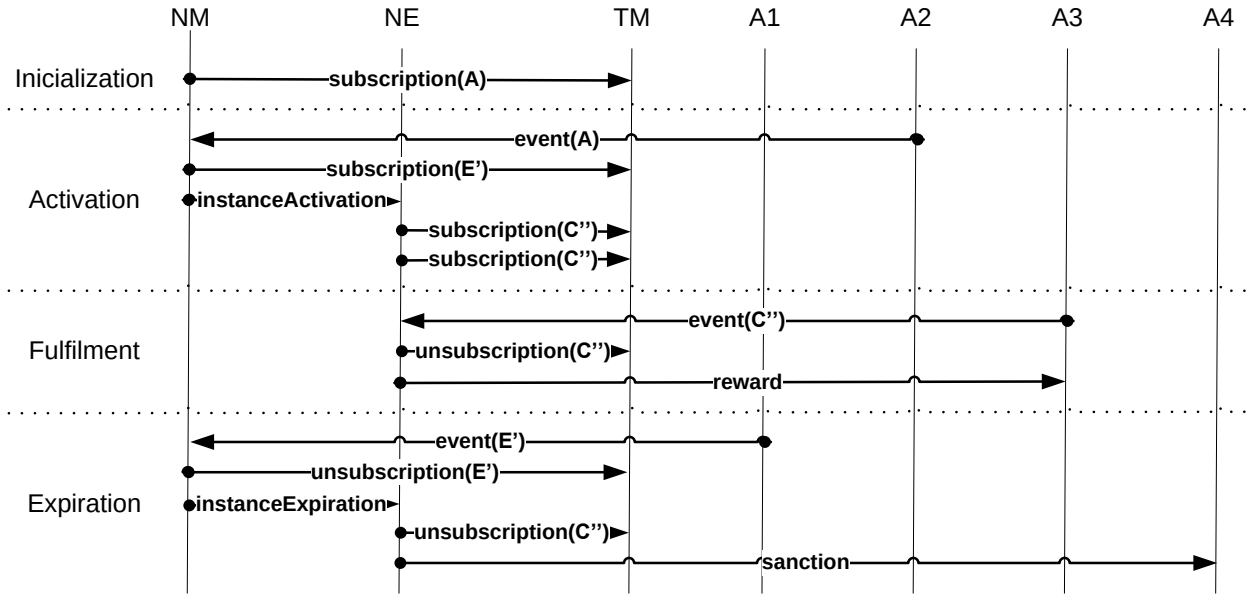

Figure 9.7: Messages exchanged in MaNEA when a single norm is controlled

the norm has been fulfilled and the NE rewards agent $A3$ and unsubscribes from event C''. As previously mentioned, the NE sends an event for informing all subscribed agents about the fulfilment of a norm. Here, we will assume that only one event is sent for rewarding the agent that has complied with the norm, as occurs in the other two proposals. Finally, when agent $A2$ sends the expiration event (Figure 9.7 messages $event(E')$), then the NM unsubscribes from this event and sends an *instanceExpiration* event to the NE. The NE unsubscribes from those events that are pending (Figure 9.7 messages $unsubscription(C'')$) and sanctions $A4$ (Figure 9.7 message *sanction*), since this agent has not complied with the norm.

According to Figure 9.7 the number of messages that are required for controlling norms in MaNEA is:

- *Initialization.* 1 message for subscribing to the activation condition;
- *Activation.* 1 message for sending the activation event, 1 message for subscribing to the expiration condition and 1 message for informing the NE about the new instance. Moreover, for each agent that is playing the target role 1 message is sent for subscribing to the norm condition. Thus, the total number of messages that are sent per activation is $3 + \theta_T^n$, where θ_T^n is the number of agents that are under the influence of norm n (i.e., the number of agents that enact role T).
- *Expiration.* When the norm expires 3 messages are sent: 1 that corresponds to the

expiration event; 1 sent by the NM for unsubscribing from this event and 1 for informing the NE about the expiration of the instance.

- *Fulfilment.* When the obliged action is performed by an agent that is affected by the norm 3 messages are sent: 1 for sending the event informing about performance of the norm condition, 1 for unsubscribing from this event and 1 for rewarding the agent.
- *Violation.* When an agent violates a norm, then the NE sends 2 messages: 1 for unsubscribing from the norm condition and 1 for sanctioning the offender agent.
- *Reporting.* No message is sent for this purpose in MaNEA.

The total number of messages required in MaNEA for controlling a single norm n:

$$\underbrace{1}_{Initialization} + \underbrace{(3+\theta_T^n)*\theta_A^n}_{Activation} + \underbrace{3*\theta_{E'}^n}_{Expiration} + \underbrace{3*\theta_F^n}_{Fulfilment} + \underbrace{2*\theta_V^n}_{Violation} + \underbrace{0}_{Reporting}$$

where $\theta_{E'}^n$ is the number of times that instances of a given norm n have expired. θ_A^n, θ_F^n and θ_V^n are the number of times that an observer detects that an instantiation of the activation holds, the number of times that a norm has been fulfilled, and the number of times that a norm has been violated, respectively.

Finally if we consider an application scenario that is controlled by a set of norms ($\mathcal{N} = \{n_1, \cdots, n_j\}$) the number of messages exchanged is:

$$\sum_{i=1}^{j}(1 + (3+\theta_T^{n_i})*\theta_A^{n_i} + 3*\theta_{E'}^{n_i} + 3*\theta_F^{n_i} + 2*\theta_V^{n_i})$$

In the three proposals the number of messages depends on several factors such as: the number of times that norms are activated, expired violated and fulfilled. As far as we know, there is not any work that analyses the occurrence of norm activations, expirations, fulfilments, violations and the number of agents that are affected by norms in average. In order to compare empirically the three proposals, we have developed a set of experiments that are described below.

9.7.2 Experimental Results

In this section we describe the set of experiments that we carried out to experimentally evaluate the performance of MaNEA with respect to Cardoso & Oliveira' approach and Modgil et al.

framework. We compute the number of messages that is required for controlling norms in each approach. Therefore, we compare the number of messages that are sent on average in each one of the three proposals.

We considered an scenario with the parameters that we sum up in Table 9.2. This scenario, we had 100 agents. These agents may enact one or more roles randomly. Specifically, 10 different roles have been considered. In order to specify the desired behaviour of these roles, 20 norms have also been created. Norms are also randomly assigned to roles. Each norm is defined in terms of three conditions, which correspond to the activation, expiration and normative condition (i.e., A, E and C). We assume that these conditions are expressed in terms of actions that agents perform or events that inform about the performance of actions (in case of MaNEA). Therefore, there are 60 ($i.e., 20*3$) actions (or *normative actions*) that are selected randomly from a set of 100 actions (60% of the actions have normative consequences). Finally, each action can be instantiated in 10 different ways. Moreover, we have performed 6 different experiments to illustrate the number of messages with respect to: the number of iterations; the number of actions; the number of norms; the number of instantiations; the number of agents; and the number of roles. In the experiments the values of the parameters range as indicated by the *Experimentation Interval* column in Table 9.2. The results of these experiments are described below.

Parameter	Fixed Value	Experimentation Interval
# of iterations	100	$[100, 1000]$
# of actions	100	$[10, 200]$
# of norms	20	$[1, 100]$
# of instantiations	10	$[1, 20]$
# of agents	100	$[0, 500]$
# of roles	10	$[1, 100]$

Table 9.2: Parameters used in the experiments

Since MaNEA is aimed to control open MAS, which are populated by heterogeneous agents; we do not want to make any assumption about the agents' capabilities to reason about norm or the agents' goals. Thus, in each iteration each agent performs an action that is randomly selected from the 1000 concrete actions (these concrete actions correspond to the 10 ways in which each one of the 100 actions can be instantiated).

9.7.2.1 Number of Iterations

Figure 9.8 illustrates the number of messages that are sent to control norms with respect to the number of iterations that the scenario has executed. As the results show, in the three proposals the number of messages increases linearly with the number of iterations. When the number of iterations increases, more actions are executed and more actions must be reported. Moreover, there are more possibilities that agents perform any of the 60 normative actions. As mentioned before, these normative actions may cause the activation, expiration, fulfilment and violation of norms. As the theoretical results illustrate, they are key factors that determine the number of messages that are sent in the three proposals. As one could expect, in the proposals in which all actions are reported (i.e., Cardoso & Oliveira' and Modgil et al. frameworks) the line has a higher slope. Moreover, the number of messages in Modgil et al. framework is slightly higher than Cardoso & Oliveira' approach[4]. We can conclude that MaNEA performs better than the other two proposals in the conditions of this experiment. In the rest of the experiments we only show the results that are obtained with 100 iterations. However, there are not significant differences among the results obtained in that experiments when the number of iterations changes.

9.7.2.2 Number of Actions

Figure 9.9 illustrates the performance of the three proposals with respect to the number of actions that can be executed by agents. As the number of actions increases the performance of MaNEA gets better than the other two proposals. This is explained by the fact that if there are more actions that can be performed by agents, the probability that an agent performs a normative action is lower. For a number of actions higher than 60, the performance of MaNEA is better than Cardoso & Oliveira' approach and Modgil et al. framework. This can be considered as a good result since there are 20 norms and, as a consequence, there are 60 normative actions. When the number of actions is less than 60 all of them are normative. Thus, all of the actions that can be performed by agents are controlled by norms and it is better to use an overhearing approach or, even better, a regimentation framework such as an EI. This

[4]As one could expect from the theoretical results:

$$(|\mathcal{A}| * I) + \sum_{i=1}^{j}(\theta_F^{n_i} + \theta_V^{n_i}) < (|\mathcal{A}| * I) + \sum_{i=1}^{j}(3 + \theta_A^{n_i} + \theta_E^{n_i} + \theta_C^{n_i} + \theta_F^{n_i} + \theta_V^{n_i})$$

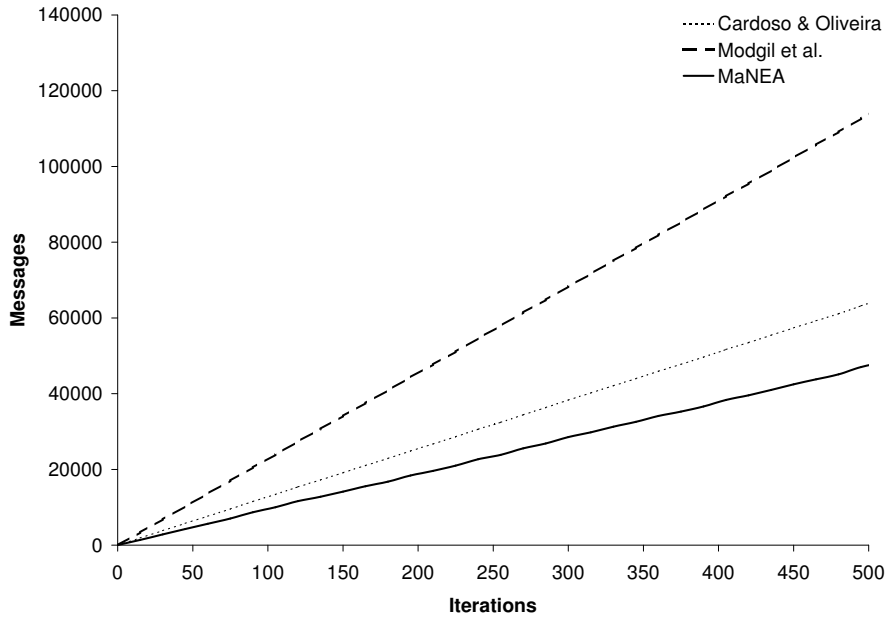

Figure 9.8: Performance of MaNEA, Cardoso & Oliveira' approach and Modgil et al. frameworks with respect to the number of iterations

also means that MaNEA performs worse when a MAS is over-regulated.

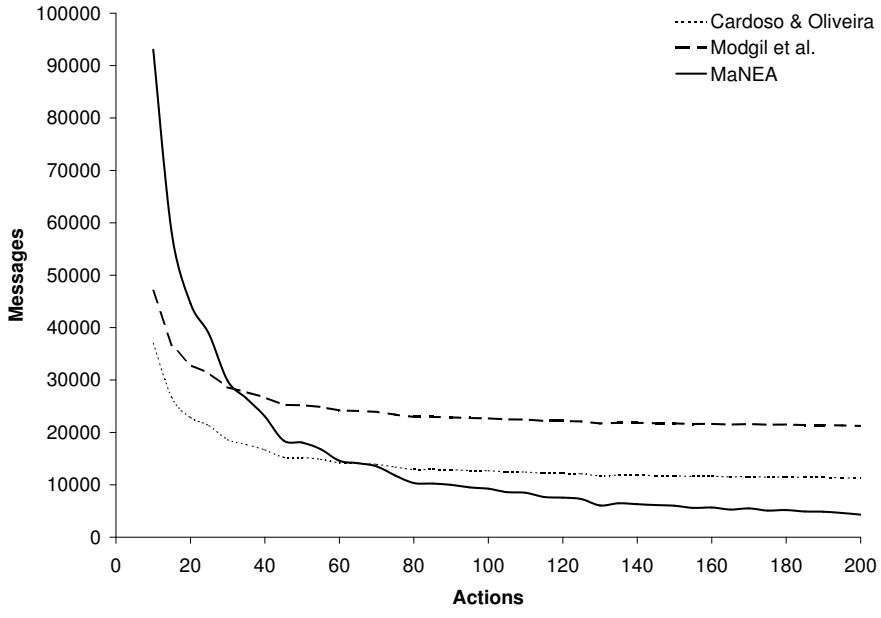

Figure 9.9: Performance of MaNEA, Cardoso & Oliveira' approach and Modgil et al. frameworks with respect to the number of actions

9.7.2.3 Number of Norms

Figure 9.10 illustrates the performance of the three proposals with respect to the number of norms. In this experiment, the number of actions increases linearly with the number of norms to maintain the ratio between the number of norms and actions[5]. In the three proposals, the number of messages that are sent remains quite stable regardless of the number of norms. In light of these results, we can conclude that MaNEA performs better than Modgil et al. framework and Cardoso & Oliveira' approach regardless of the number of norms that are controlled.

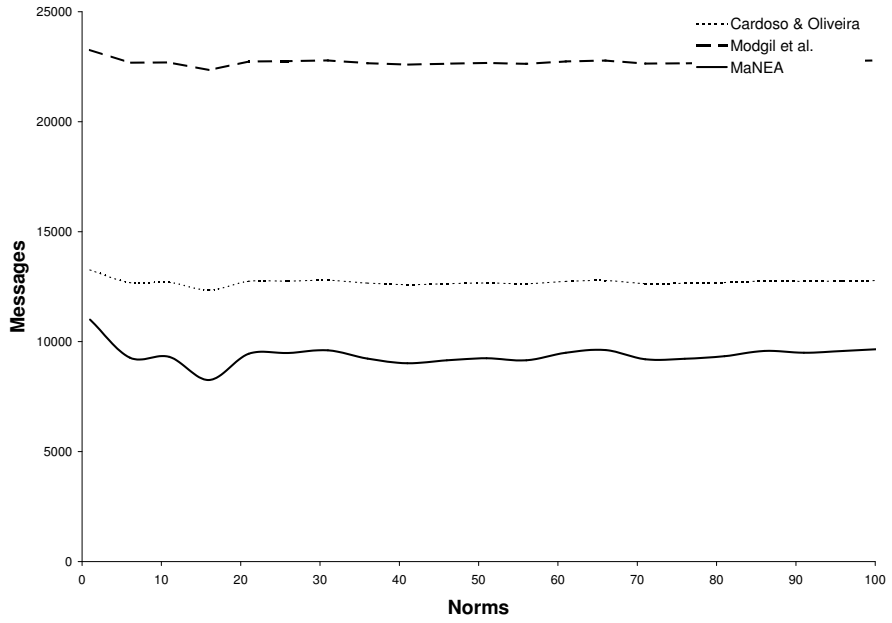

Figure 9.10: Performance of MaNEA, Cardoso & Oliveira' approach and Modgil et al. frameworks with respect to the number of norms

9.7.2.4 Number of Instantiations

Figure 9.11 illustrates the performance of the three proposals with respect to the number of ways in which each action can be instantiated. As the number of instantiations increases the performance of MaNEA gets better than the other two proposals. Thus it is more scalable for an increasing number of instantiations. If actions can be instantiated in more ways, the probability that an agent performs the concrete instance that causes the fulfilment, violation or expiration of an instance is lower. For a number of instantiations higher than 7 the performance

[5]If there are more norms than actions, then all actions are controlled by norms. As previously mentioned, in this case it is better to use a regimentation system.

of MaNEA is better than Cardoso & Oliveira' approach and Modgil et al. frameworks. This can be considered as a good result since there are 100 agents and it seems reasonable that these 100 agents are able to execute each action in 7 different ways.

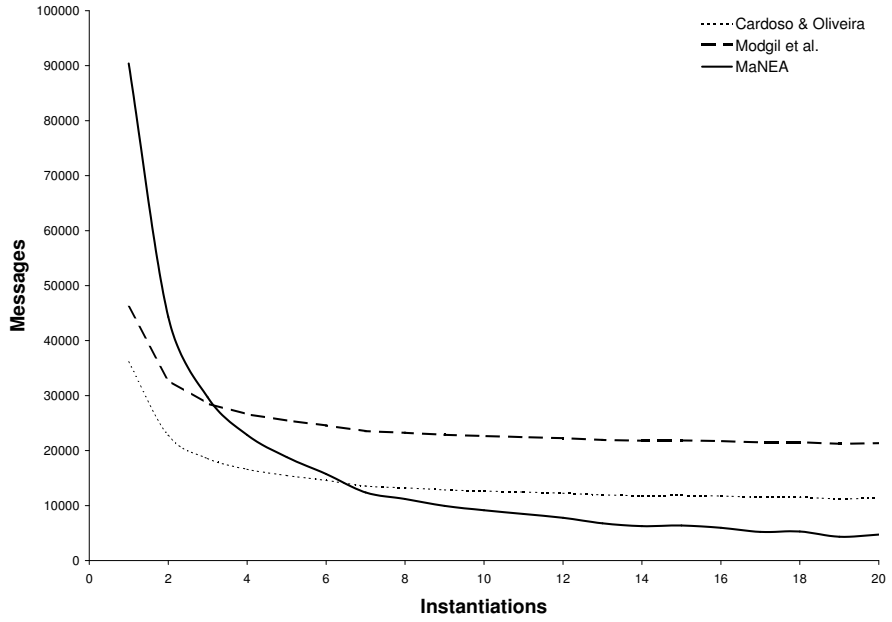

Figure 9.11: Performance of MaNEA, Cardoso & Oliveira' approach and Modgil et al. frameworks with respect to the number of instantiations

9.7.2.5 Number of Agents

Figure 9.12 illustrates the performance of the three proposals with respect to the number of agents. In this experiment, the number of instantiations also increases linearly with the number of agents to maintain the ratio between the number of agents and instantiations. It makes sense to assume that if there are more agents there will be more different kinds of agents that will be able to execute actions in more different ways. In the three proposals the number of messages increases linearly with the number of agents; which is consistent with the theoretical results previously explained. If there are more agents there are more actions to be reported. In light of these results we can conclude that the MaNEA performs better than Cardoso & Oliveira' approach and Modgil et al. frameworks regardless of the number of agents.

9.7.2.6 Number of Roles

Figure 9.13 illustrates the performance of the three proposals with respect to the number of roles. In the three proposals, as the number of roles increases, fewer messages are sent. Reasons

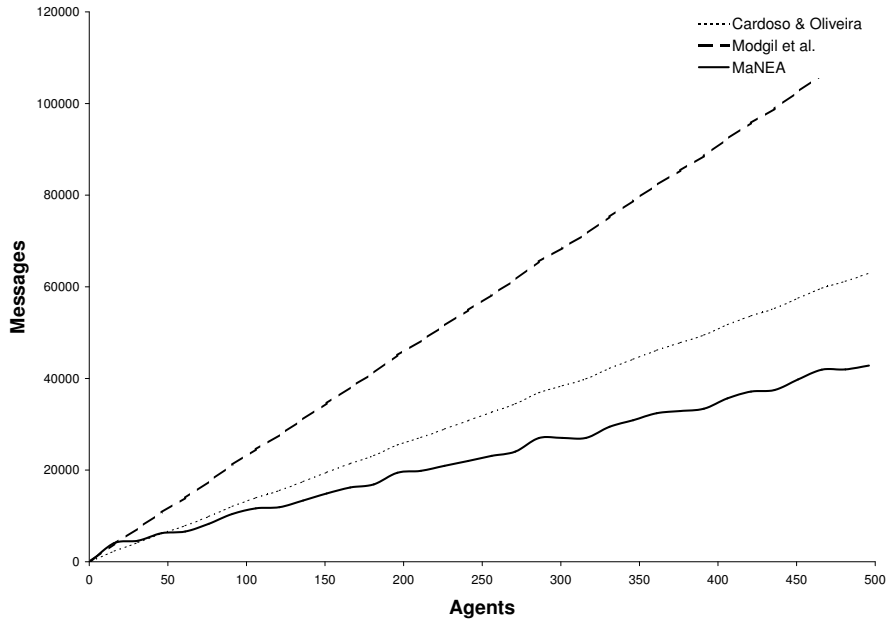

Figure 9.12: Performance of MaNEA, Cardoso & Oliveira' approach and Modgil et al. frameworks with respect to the number of agents

beyond this decrement are related to the fact that if there are a higher number of roles, each role will be affected by fewer norms and also fewer agents will be affected by them. Therefore, there is a lower probability of norm fulfilment and violation. For a number of roles higher than 3, which seems reasonable considering the number of agents and norms, MaNEA performs better than Modgil et al. framework.

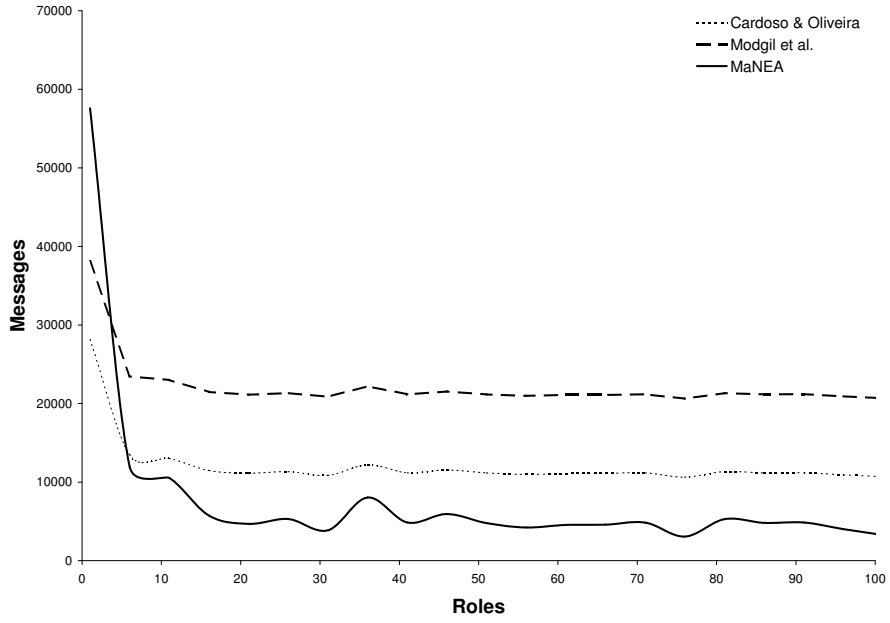

Figure 9.13: Performance of MaNEA, Cardoso & Oliveira' approach and Modgil et al. frameworks with respect to the number of roles

In general, the performance of MaNEA, which has been measured in terms of the number of messages, is better than centralized and distributed proposals with the same capabilities. MaNEA takes as a reference a trace event system based on a publish/subscription procedure. In light of the results described above, we have demonstrated that under certain circumstances the use of a tracing service implies an outstanding reduction of the number of messages exchanged for controlling norms.

9.8 Contributions

The main aim of MaNEA is to overcome problems of existing proposals on norm enforcement. Thus, the requirements taken into account by our proposal are:

- **Automatic Enforcement.** Our proposal enforces norms providing support to those agents that are not endowed with normative reasoning capabilities. In addition, the generation of events for informing about sanctions and rewards allows norm-aware agents to use this information for selecting the most suitable interaction partners.

- **Control of general norms.** Our definition of norm is based on the notion of event. Thereby, norms are defined in terms of events that can be generated independently by different tracing entities. These events may be: *generic* events that represent application independent information; and *application* events that are domain dependent information.

- **Dynamic**. Magentix2 allows the dynamic modification of norms. Moreover, new event types can be dynamically defined at runtime. Accordingly, our proposal has been designed taking into account the possibility that norms and events can be created or deleted online. Moreover, MaNEA is endowed with mechanisms to control the dynamic enactment of roles.

- **Efficient, Distributed and Robust**. Finally, MaNEA is built upon a trace event system, which provides support for indirect communication in a more efficient way than overhearing approaches. In general situations, the use of a tracing service reduces the number of messages required to control norms. In MaNEA the reasoning about norm enforcement is distributed and performed in the two layers, which reduces the computational cost of the algorithms executed to reason about norm enforcement. Besides that, we

have provided a preliminary solution to the adaptation of the architecture in response to situations in which the number of agents or norms to be controlled changes dramatically.

9.9 Conclusions

In this chapter, we have described a Norm-Enforcing Architecture (MaNEA) that has been developed considering the facilities provided by the Magentix2 platform. This architecture is responsible for monitoring and enforcing the norms that regulate VOs. A prototype of the n-BDI architecture has been developed in Jason. Since Magentix2 provides native support for executing Jason agents, thus this implementation can be used to implement norm-autonomous agents capable of participating in VOs that are controlled by norms. As previously mentioned, several simplifications have been made in this prototype. As future work we plan to extend this prototype to include al the functionalities provided by the n-BDI agent architecture. The next chapter, presents the main contributions of this thesis and points out future lines of research.

Chapter 10

Conclusions

As mentioned in the introduction of this thesis, the main objective of this thesis is to develop norm reasoning mechanisms suitable for open MAS. Specifically, this thesis is aimed at developing both an agent architecture, which allows agents to reason autonomously about norms; and a norm-enforcing architecture, which allows norms to be controlled in open MAS.

10.1 Contributions

The n-BDI architecture proposed in this thesis models norm-autonomous agents endowed with all the norm-reasoning capabilities. According to the features of the n-BDI architecture, there may be different reasons why a n-BDI agent may violate a given norm: i) Since it does not *know* the norm; i.e., the agent has not been informed about the existence of that norm. ii) Since it does not *accept* the norm, i.e., the norm is not salient enough and the agent decides not to follow it. iii) Since it does not consider the norm as *relevant* to its situation; i.e., it considers that the norm is not active according to its uncertain knowledge of the world or the agent does not believe that it is under the norm scope. iv) Since it is not willing to *comply* with the norm. In the n-BDI architecture the norm compliance decisions can be justified by rational motivations, which are related to self-interest and expectations; and non-rational reasons, which are related to emotions. v) Since it does not consider the norm as *coherent* to its mental state. Therefore, an agent may decide to violate a norm because it is in conflict with other relevant cognitive elements (e.g., a norm that is incompatible with an internal desire which has more priority). vi) Since it is not *capable* of fulfilling the norm. This issue is more related to the decision making procedure and is beyond the scope of this work. As far as we are concerned, none of

the existing proposals on norm-autonomous agents allow designers to model agents that exhibit a behaviour in which norms can be violated according to all of the above mentioned reasons.

Moreover, this thesis proposes a norm-enforcing architecture that allows norms to be controlled considering the facilities provided by the Magentix2 platform. Our proposal enforces norms providing support to agents that are not endowed with normative reasoning capabilities and norm-aware agents. Moreover, it has been designed taking into account the possibility that norms and events can be created or deleted on-line. Finally, our norm-enforcing architecture is built upon a trace event system, which provides support for indirect communication in a more efficient way than overhearing approaches. Thus, the use of a tracing service reduces the number of messages required to control norms. Besides that, we have provided a preliminary solution to the adaptation of the architecture in response to situations in which the number of agents or norms to be controlled changes dramatically.

Finally a prototype of the n-BDI architecture has been implemented in Jason. Thereby, the n-BDI architecture can be used to develop norm-autonomous agents that participate in Open MAS developed in Magentix2.

10.2 Future Works

In this section, we outline some of the most challenging possible future directions in the research field of norms and MAS. These possible directions are open challenges identified during the realization of this thesis.

- *Evolution of decisions about norm compliance.* One of the problems that has not been considered by the n-BDI proposal is the evolution of the decisions about norm compliance. The decisions about norm compliance are quite unstable and may change several times along the agent life. This is due to the fact that the decisions about norm compliance are not considered for updating the salience of norms. In other words, any time that an agent observes that a norm has been fulfilled, then the salience of this norm must be updated. Therefore, the norm becomes more important and agents will be more willing to comply with it.

- *Norm-enforcer agents.* The great majority of the works on norm enforcement are based on the existence of a shared reality which is fully observed. However, this assumption

of fully observability is too much strong in dynamic and uncertain domains. In this sense, the detection and reaction to norm violations should be carried out according to a partial observation of the real word. As future work, we plan to deal with complex scenarios in which there are norms whose violation cannot be directly observed, since they regulate situations that take place out of the institution boundaries. Or even more, norms that can be interpreted ambiguously. This entails the development of intelligent and proactive norm-enforcing entities (i.e., agents) [CAB11d] capable of learning new norms dynamically and deliberating about norm enforcement given that there is a partial and uncertain observability of both the world and the agent interactions. Specifically, the n-BDI agent architecture can be extended with norm enforcement capabilities. Thus, agents would be provided with mechanisms that allow them to evaluate partners according to norms and performing sanctioning and rewarding actions in response.

- *Dispute resolution.* In the existing literature, the solutions to the norm compliance problem assume that norms are unambiguously interpreted. Thus, norm violations are detected by analysing illocutions and actions performed by agents, which are fully observed by an institution. However, deciding whether or not a norm has been violated is a matter that should be agreed on if we consider the implications that uncertain environments have. On the one hand, uncertainty entails that agents cannot assume their beliefs as immovable. On the other hand, the existence of private interactions implies that relevant facts may be unknown to other agents which are not directly involved in these interactions. Finally, environments are populated by heterogeneous agents which might give different interpretations to norms. In this situation there might be inconsistencies among the reality perceived by agents. Consequently, norm enforcing agents require capabilities for reaching a consensus about norm compliance, defining which agents are responsible and determining the repairing actions.

- *Collective decisions about norm compliance.* For the moment being, n-BDI agents make decisions individually. However, there may be scenarios in which norms can only be fulfilled as a result of the cooperation among several agents. Hence, agents must take decisions about norm compliance collectively. This entails some other issues treated in MAS and agreement technologies, such as coordination, cooperation, delegation, etc.

10.3 Related Publications

Next, all publications describing the results of this thesis are listed.

10.3.1 Publications in Journals

Journals Indexed by the SCI

[CAB11c] N. Criado, E. Argente and V. Botti

Open Issues for Normative Multi-Agent Systems

AI Communications.24(3) pp: 233–264. (2011).

Impact Factor 0.837 (Q3 COMPUTER SCIENCE, ARTIFICIAL INTELLIGENCE)

[CAB11a] N. Criado, E. Argente and V. Botti

THOMAS: An Agent Platform For Supporting Normative Multi-Agent Systems

Journal of Logic and Computation. Available on-line 2011.

Impact Factor 0.586 (Q4 COMPUTER SCIENCE, THEORY & METHODS)

Other Journals

[HCAJ09b] S. Heras, N. Criado, E. Argente and V. Julian

Norm Emergency through Argumentation

Journal of Physical Agents Vol. 3 N. 3 pp. 31-38. (2009)

[CAJB08] N. Criado, E. Argente, V. Julian and V. Botti

Servicios Organizacionales Para La Plataforma De Agentes Spade

IEEE Latin America Transactions Vol. 6 N. 6 pp. 550-555. (2008)

10.3.2 Publications in Conferences

Conferences Indexed by the CORE[1]

[CANB12b] N. Criado, E. Argente, P. Noriega and V. Botti.

Determining the Willingness to Comply With Norms. 11th Int. Conf. on Autonomous Agents and Multiagent Systems (AAMAS 2012) pp. In press. (2012). CORE A

[1] http://www.core.edu.au/

[**Cri11**] N. Criado.

Reasoning About Norms Within Uncertain Environments

10th Int. Conf. on Autonomous Agents and Multiagent Systems (AAMAS 2011) pp. 1331-1332. (2011). CORE A

[**CABN11**] N. Criado, E. Argente, V. Botti and P. Noriega

Reasoning About Norm Compliance

10th Int. Conf. on Autonomous Agents and Multiagent Systems (AAMAS 2011) pp. 1191-1192. (2011). CORE A

[**dVCC$^+$10**] E. del Val, N. Criado, C. Carrascosa, V. Julian, M. Rebollo, E. Argente and V. Botti

THOMAS: A Service-Oriented Framework For Virtual Organizations

9th Int. Conf. on Autonomous Agents and Multiagent Systems (AAMAS 2010) pp. 1631-1632. (2010). CORE A

[**CAB10a**] N. Criado, E. Argente and V. Botti

A BDI Architecture for Normative Decision Making

9th Int. Conf. on Autonomous Agents and Multiagent Systems (AAMAS 2010) pp. 1383-1384. (2010). CORE A

[**CAB10b**] N. Criado, E. Argente and V. Botti

Normative Deliberation in Graded BDI Agents

8th German Conference on Multi-Agent System Technologies (MATES-10) Vol. 6251 pp. 52-63. (2010). CORE B

[**VCRA09**] E. Del Val, N. Criado, M. Rebollo and E. Argente

Normative Time-Bounded Service Logic

7th European Workshop on Multi-Agent Systems (EUMAS'09) pp. 1-13. (2009). CORE C

[**HCAJ09a**] S. Heras, N. Criado, E. Argente and V. Julian

A Dialogue-Game Approach for Norm-based MAS Coordination

International Conference on Hybrid Artificial Intelligence Systems (HAIS). Vol. 1 N. 5572 pp. 468-475. (2009). CORE C

[**ACBJ08**] E. Argente, N. Criado, V. Botti and V. Julian

Norms for Agent Service Controlling

European Workshop on Multi-Agent Systems (EUMAS) pp. 1-15. (2008). CORE C

Conferences Indexed by the CSCR

[**CAB11d**] N. Criado, E. Argente and V. Botti

Towards Norm Enforcer Agents

9th International Conference on Practical Applications of Agents and Multi-Agent Systems (PAAMS). In Press. (2011). CSCR: 0.56 (position 51 / 701)

[**CHAJ10b**] N. Criado, S. Heras, E. Argente and V. Julian

Normative Argumentation

8th International Conference on Practical Applications of Agents and Multi-Agent Systems (PAAMS) Vol. 71 pp. 29-36. (2010). CSCR: 0.56 (position 51 / 701)

[**CVRA09**] N. Criado, E. Del Val, M. Rebollo and E. Argente

A Logic for Normative Time-Bounded Services

Conferencia de la Asociacin Espaola para la Inteligencia Artificial (CAEPIA) pp. 645-654. (2009). CSCR 0.55 (position 54 / 701)

[**CAB09a**] N. Criado, V.Botti, E. Argente

A Normative Model For Open Agent Organizations

International Conference on Artificial Intelligence (ICAI) Vol. 1 pp. 101-107. (2009). CSCR: 0.8 (position 18 / 701)

[**dVCR$^+$09**] E. del Val, N. Criado, M. Rebollo, E. Argente, V. Julian

Service-Oriented Framework for Virtual Organizations

International Conference on Artificial Intelligence (ICAI) Vol. 1 pp. 108-114. (2009). CSCR: 0.8 (position 18 / 701)

[**CAJB09**] N. Criado, E. Argente, V. Julian and V. Botti

Designing Virtual Organizations

International Conference on Practical Applications of Agents and Multi-Agent Systems (PAAMS) Vol. 55 pp. 319-328. (2009). CSCR: 0.56 (position 51 / 701)

[CJA09] N. Criado, V. Julian and E. Argente

Towards the Implementation of a Normative Reasoning Process

International Conference on Practical Applications of Agents and Multi-Agent Systems (PAAMS) Vol. 55 pp. 440-449. (2009). CSCR: 0.56 (position 51 / 701)

[CAJB07] N. Criado, E. Argente, V. Julian and V. Botti

Organizational Services for SPADE agent platform

Workshop Internacional sobre Aplicaciones Prcticas de Agentes y Sistemas Multiagente (IWPAAMS) Vol. 1 pp. 31-40. (2007). CSCR: 0.56 (position 51 / 701)

Other Conferences

[CAG+10] N. Criado, E. Argente, A. Garrido, J. A. Gimeno, F. Igual, V. Botti, P. Noriega and A. Giret

Norm enforceability in Electronic Institutions?

11th International Workshop on Coordination, Organization, Institutions and Norms in Multi-Agent Systems (COIN@MALLOW2010) pp. 49-64. (2010)

[CANB10] N. Criado, E. Argente, P. Noriega and V. Botti

Towards a Normative BDI Architecture for Norm Compliance

11th International Workshop on Coordination, Organization, Institutions and Norms in Multi-Agent Systems (COIN@MALLOW2010) pp. 65-81. (2010)

[CAB10c] N. Criado, E. Argente and V. Botti

Rational Strategies for Autonomous Norm Adoption

9th International Workshop on Coordination, Organization, Institutions and Norms in Multi-Agent Systems (COIN@AAMAS2010) pp. 9-16. (2010)

[CHAJ10a] N. Criado, S. Heras, E. Argente and V. Julian

Contract Argumentation in Virtual Organizations

5th International Workshop on Normative Multi-Agent Systems, (NORMAS-10) pp. 55-59. (2010)

[CJBA09] N. Criado, V. Julian, V. Botti and E. Argente

A Norm-based Organization Management System

International Workshop Coordination, Organizations, Institutions and Norms (COIN). Pp. 1-16.(2009)

[**ACJB08**] E. Argente, N. Criado, V. Julian and V. Botti

Designing Norms in Virtual Organizations

Congrs Internacional de l'Associaci Catalana d'Intelligncia Artificial (CCIA) Vol. 184 pp. 16-23. (2008)

10.3.3 Book Chapters

[**CANB12a**] N. Criado and E. Argente and P. Noriega and V. Botti

A Distributed Architecture for Enforcing Norms in Open MAS

Advanced Agent Technology. Vol. 7068. pp. 457-471. Springer. (2012)

[**CAB11b**] N. Criado, E. Argente and V. Botti

Rational Strategies for Norm Compliance in the n-BDI Proposal

Coordination, Organizations, Institutions, and Norms in Agent Systems VI. Vol. 6541. pp. In Press. Springer. (2011)

[**CAG$^+$11**] N. Criado, E. Argente, A. Garrido, J. A. Gimeno, F. Igual, V. Botti, P. Noriega, A. Giret

Norm enforceability in Electronic Institutions?

Coordination, Organizations, Institutions, and Norms in Agent Systems VI. Vol. 6541. pp. In Press. Springer. (2011)

[**CJBA10**] N. Criado, V. Julian, V. Botti and E. Argente

A Norm-based Organization Management System

Coordination, Organizations, Institutions, and Norms in Agent Systems V. Vol. 6069. pp. 19-35. Springer. (2010)

Bibliography

[AA07] M. Anderson and S.L. Anderson. Machine ethics: Creating an ethical intelligent agent. *The AI Magazine*, 28(4):15–26, 2007.

[AÁNDVS10] H. Aldewereld, S. Álvarez-Napagao, F.P.M. Dignum, and J. Vázquez-Salceda. Making Norms Concrete. In *Proc. of the International Conference on Autonomous Agents and Multiagent Systems (AAMAS)*, pages 807–814, 2010.

[AB71] C. E. Alchourrn and E. Bulygin. *Normative Systems*. Springer, 1971.

[ABC+11] E. Argente, V. Botti, C. Carrascosa, A. Giret, V. Julian, and M. Rebollo. An Abstract Architecture for Virtual Organizations: The THOMAS approach. *Knowledge and Information Systems*, pages 1–35, 2011.

[ABJ11] E. Argente, V. Botti, and V. Julian. Gormas: An organizational-oriented methodological guideline for open MAS. *Agent-Oriented Software Engineering X*, pages 32–47, 2011.

[ACBJ08] E. Argente, N. Criado, V. Botti, and V. Julian. Norms for agent service controlling. In *Proc. of the European Workshop on Multi-agent Systems (EUMAS)*, pages 1–15, 2008.

[ACCC08] G. Andrighetto, M. Campennì, F. Cecconi, and R. Conte. How agents find out norms: A simulation based model of norm innovation. In *Proc. of the International Workshop on Normative Multiagent Systems (NORMAS)*, pages 16–30, 2008.

[ACCP07] G. Andrighetto, M. Campenni, R. Conte, and M. Paolucci. On the immergence of norms: a normative agent architecture. In *Proc. of AAAI Symposium, Social and Organizational Aspects of Intelligence*, 2007.

[ACJB08] E. Argente, N. Criado, V. Julian, and V. Botti. Designing norms in virtual organizations. In *Proc. of the International Conference of the Catalan Association for Artificial Intelligence (CCIA)*, volume 184, pages 16–23. IOS Press, 2008.

[AdBD10] L. Astefanoaei, F.S. de Boer, and M. Dastani. Strategic Executions of Choreographed Timed Normative Multi-Agent Systems. In *Proc. of the International Conference on Autonomous Agents and Multiagent Systems (AAMAS)*, pages 965–972, 2010.

[ADM07] H. Aldewereld, F.P.M. Dignum, and J-J.Ch. Meyer. From norms to interaction patterns: Deriving protocols for agent institutions. In *Proc. of the Workshop on Programming Multi-Agent Systems (ProMAS)*, pages 22–37, 2007.

[AGM85] C.E. Alchourrón, P. Gärdenfors, and D. Makinson. On the logic of theory change: Partial meet contraction and revision functions. *Journal of symbolic logic*, 50(2):510–530, 1985.

[AHK02] R. Alur, T.A. Henzinger, and O. Kupferman. Alternating-time temporal logic. *Journal of the ACM (JACM)*, 49(5):672–713, 2002.

[Ald09] H. Aldewereld. Autonomy vs. conformity: an institutional perspective on norms and protocols. *The Knowledge Engineering Review*, 24(4):410–411, 2009.

[And58] A.R. Anderson. A reduction of deontic logic to alethic modal logic. *Mind*, 67:100–103, 1958.

[AP01] A. Artikis and J. Pitt. A formal model of open agent societies. In *Proc. of the International Conference on Autonomous Agents and Multiagent Systems (AAMAS)*, pages 192–193, 2001.

[ASP09] A. Artikis, M. Sergot, and J. Pitt. Specifying norm-governed computational societies. *ACM Transactions on Computational Logic (TOCL)*, 10(1):1, 2009.

[AVC10] G. Andrighetto, D. Villatoro, and R. Conte. Norm internalization in artificial societies. *AI Communications*, 23(4):325–339, 2010.

[ÅvdHRA+07] T. Ågotnes, W. van der Hoek, J.A. Rodriguez-Aguilar, C. Sierra, and M. Wooldridge. On the logic of normative systems. In *Proc. of the Inter-*

national Joint Conference on Artificial Intelligence (IJCAI), pages 1175–1180, 2007.

[ÅvdHW07] T. Ågotnes, W. van der Hoek, and M. Wooldridge. Normative system games. In *Proc. oh the International Conference on Autonomous Agents and Multiagent Systems (AAMAS)*, pages 129–135, 2007.

[ÅvdHW08] T. Ågotnes, W. van der Hoek, and M. Wooldridge. Robust normative systems. In *Proc. of the International Conference on Autonomous Agents and Multiagent Systems (AAMAS)*, pages 747–754, 2008.

[Bal09] T. Balke. A taxonomy for ensuring institutional compliance in utility computing. In *Proc. of the International Workshop on Normative Multiagent Systems (NORMAS)*, 2009.

[Ban91] A. Bandura. Social cognitive theory of moral thought and action. *Handbook of moral behavior and development*, 1:45–103, 1991.

[BBC+00] P. Baldi, S. Brunak, Y. Chauvin, C.A.F. Andersen, and H. Nielsen. Assessing the accuracy of prediction algorithms for classification: an overview. *Bioinformatics*, 16(5):412, 2000.

[BBT08] Guido Boella, Jan Broersen, and Leendert Torre. Reasoning about constitutive norms, counts-as conditionals, institutions, deadlines and violations. In *Proceedings of the Pacific Rim International Conference on Multi-Agents (PRIMA)*, pages 86–97, 2008.

[BC10] C. Bicchieri and A. Chavez. Behaving as expected: Public information and fairness norms. *Journal of Behavioral Decision Making*, 23(2):161–178, 2010.

[BDC+95] J.M. Bradshaw, S. Dutfield, B. Carpenter, R. Jeffers, and T. Robinson. KAoS: A generic agent architecture for aerospace applications. In *Proc. of Workshop on Intelligent Information Agents on the Conference on Information and Knowledge Management*, 1995.

[BDH+01] J. Broersen, M. Dastani, J. Hulstijn, Z. Huang, and L. van der Torre. The boid architecture – conflicts between beliefs, obligations, intentions and desires. In

Proc. of the International Conference on Autonomous Agents and Multiagent Systems (AAMAS), pages 9–16, 2001.

[Ber85] J.O. Berger. *Statistical decision theory and Bayesian analysis*. Springer, 1985.

[BGFJT11] L.A. Burdalo, A. Garcia-Fornes, V. Julian, and A. Terrasa. TRAMMAS: A tracing model for multiagent systems. *Engineering Applications of Artificial Intelligence*, 24(7):1110–1119, 2011.

[BGRvdT10] G. Boella, G. Governatori, A. Rotolo, and L. van der Torre. Lex minus dixit quam voluit, lex magis dixit quam voluit: A formal study on legal compliance and interpretation. *AI Approaches to the Complexity of Legal Systems. Complex Systems, the Semantic Web, Ontologies, Argumentation, and Dialogue*, pages 162–183, 2010.

[BHW08] R.H. Bordini, J.F. Hübner, and M. Wooldridge. *Programming multi-agent systems in AgentSpeak using Jason*, volume 8. Wiley-Interscience, 2008.

[Bic06] C. Bicchieri. *The grammar of society: The nature and dynamics of social norms*. Cambridge Univ Pr, 2006.

[BL01] G. Boella and L. Lesmo. Deliberate normative agents. In *Social order in MAS*. Kluwer, 2001.

[Blo96] I. Bloch. Information combination operators for data fusion: A comparative review with classification. *IEEE Transactions on Systems, Man and Cybernetics, Part A: Systems and Humans*, 26(1):52–67, 1996.

[BMMCd04] J. Bentahar, B. Moulin, J-J.Ch. Meyer, and B. Chaib-draa. A modal semantics for an argumentation-based pragmatics for agent communication. In *Proc. International Workshop on Argumentation in Multi-Agent Systems (ArgMAS)*, pages 44–63, 2004.

[BPvdT09a] G. Boella, G. Pigozzi, and L. van der Torre. Normative framework for normative system change. In *Proc. of the International Conference on Autonomous Agents and Multiagent Systems (AAMAS)*, volume 1, pages 169–176, 2009.

[BPvdT09b] G. Boella, G. Pigozzi, and L. van der Torre. Normative systems in computer science - ten guidelines for normative multiagent systems. In *Proc. of the International Workshop on Normative Multiagent Systems (NORMAS)*, 2009.

[BUJ+03] J. Bradshaw, A. Uszok, R. Jeffers, N. Suri, P. Hayes, M. Burstein, A. Acquisti, B. Benyo, M. Breedy, M. Carvalho, et al. Representation and reasoning for DAML-based policy and domain services in KAoS and Nomads. In *Proc. of the International Conference on Autonomous Agents and Multiagent Systems (AAMAS)*, pages 835–842, 2003.

[BvdT03] G. Boella and L. van der Torre. Permissions and obligations in hierarchical normative systems. In *Proc. of the international conference on Artificial intelligence and law (ICAIL)*, pages 109–118, 2003.

[BvdT04a] G. Boella and L. van der Torre. Regulative and constitutive norms in normative multiagent systems. In *Proc. of the International Conference on Principles of Knowledge Representation and Reasoning (KR)*, pages 255–265, 2004.

[BvdT04b] G. Boella and L. van der Torre. The social delegation cycle. In *Proc. of the International Workshop on Deontic Logic in Computer Science (DEON)*, pages 29–42, 2004.

[BvdT05a] G. Boella and L. van der Torre. Enforceable social laws. In *Proc. of the International Conference on Autonomous Agents and Multiagent Systems (AAMAS)*, pages 682 – 689, 2005.

[BvdT05b] G. Boella and L. W. N. van der Torre. Constitutive norms in the design of normative multiagent systems. In *Computational Logic in Multi-Agent Systems VI*, volume 3900 of *Lecture Notes in Computer Science*, pages 303–319, 2005.

[BvdT06] G. Boella and L. van der Torre. Game-theoretic foundations for norms. *Proc. of Artificial Intelligence Studies*, 3(26):39–51, 2006.

[BvdT08] G. Boella and L. van der Torre. Substantive and procedural norms in normative multiagent systems. *Journal of Applied Logic*, 6(2):152–171, 2008.

[BvdTV06] G. Boella, L. van der Torre, and Harko Verhagen. Introduction to normative multiagent systems. *Comput. Math. Organ. Theory*, 12(2-3):71–79, 2006.

[BvdTV07] G. Boella, L. van der Torre, and H. Verhagen. Introduction to normative multiagent systems. In *Normative Multi-agent Systems*, 2007.

[BvdTV08a] G. Boella, L. van der Torre, and H. Verhagen. Introduction to the special issue on normative multiagent systems. *Autonomous Agents and Multi-Agent Systems*, 17:1–10, 2008.

[BvdTV08b] G. Boella, L. van der Torre, and H. Verhagen. Ten challenges for normative multiagent systems. In *Proc. of the International Workshop on Normative Multiagent Systems (NORMAS)*, 2008.

[CAB09a] N. Criado, E. Argente, and V. Botti. A Normative Model For Open Agent Organizations. In *Proc. of the International Conference on Artificial Intelligence (ICAI)*, volume 1, pages 101–107, 2009.

[CAB09b] N. Criado, E. Argente, and V. Botti. A normative model for open agent organizations. In *Proc. of the International Conference on Artificial Intelligence (ICAI)*, pages 101–108. CSREA Press, 2009.

[CAB10a] N. Criado, E. Argente, and V. Botti. A BDI Architecture for Normative Decision Making (Extended Abstract). In *Proc. of the Interntaional Confference on Autonomous Agents and Multiagent Systems (AAMAS)*, pages 1383–1384, 2010.

[CAB10b] N. Criado, E. Argente, and V. Botti. Normative Deliberation in Graded BDI Agents. In *Proc. of the German Conference on Multi-Agent System Technologies (MATES)*, pages 52–63, 2010.

[CAB10c] N. Criado, E. Argente, and V. Botti. Rational Strategies for Autonomous Norm Adoption. In *Proc. of the International Workshop on Coordination, Organization, Institutions and Norms in Agent Systems (COIN)*, pages 9–16, 2010.

[CAB11a] N. Criado, E. Argente, and V. Botti. Open Issues for Normative Multi-Agent Systems. *AI Communications*, 24(3):233–264, 2011.

[CAB11b] N. Criado, E. Argente, and V. Botti. Rational Strategies for Norm Compliance in the n-BDI Proposal. In *Coordination, Organizations, Institutions, and Norms in Agent Systems VI*, pages 1–20. 2011.

[CAB11c] N. Criado, E. Argente, and V. Botti. THOMAS: An Agent Platform For Supporting Normative Multi-Agent Systems. *Journal of Logic and Computation*, 2011.

[CAB11d] N. Criado, E. Argente, and V. Botti. Towards Norm Enforcer Agents. In *Proc. of the International Conference on Practical Applications of Agents and Multi- Agent Systems (PAAMS)*, volume 89 of *Advances in Intelligent and Soft Computing*, pages 135–142, 2011.

[CABN11] N. Criado, E. Argente, V. Botti, and P. Noriega. Reasoning about norm compliance (extended abstract). In *Proc. of the International Conference on Autonomous Agents and Multiagent Systems (AAMAS)*, pages 1191–1192, 2011.

[CAC10] R. Conte, G. Andrighetto, and M. Campenní. On norm internalization. a position paper. In *Proc. of the European Workshop on Multi-agent Systems (EUMAS)*, pages 1–13, 2010.

[CACC09] M. Campenní, G. Andrighetto, F. Cecconi, and R. Conte. Normal= normative? the role of intelligent agents in norm innovation. *Mind & Society*, 8(2):153–172, 2009.

[CACP07] R. Conte, G. Andrighetto, M. Campennì, and M. Paolucci. Emergent and immergent effects in complex social systems. In *Proc. of the AAAI Symposium, Social and Organizational Aspects of Intelligence*, pages 42–47, 2007.

[CAG+10] N. Criado, E. Argente, A. Garrido, J. A. Gimeno, F. Igual, V. Botti, P. Noriega, and A. Giret. Norm enforceability in Electronic Institutions? In *Proc. of the International Workshop on Coordination, Organization, Institutions and Norms in Agent Systems (COIN)*, pages 49–64, 2010.

[CAG+11] N. Criado, E. Argente, A. Garrido, J. A. Gimeno, F. Igual, V. Botti, P. Noriega, and A. Giret. Norm enforceability in Electronic Institutions? In *Coordination, Organizations, Institutions, and Norms in Agent Systems VI*, pages 250–267. 2011.

[CAJB07] N. Criado, E. Argente, V. Julian, and V. Botti. Organizational Services for SPADE agent platform. In *Proc. of the Internacional Workshop on Practical*

Applications of Agents and Multiagents Systems (IWPAAMS), volume 1, pages 31–40, 2007.

[CAJB08] N. Criado, E. Argente, V. Julian, and V. Botti. Servicios organizacionales para la plataforma de agentes spade. *IEEE Latin America Transactions*, 6:550–555, 2008.

[CAJB09] N. Criado, E. Argente, V. Julian, and V. Botti. Designing virtual organizations. In *Proc. of the International Conference on Practical Applications of Agents and Multi- Agent Systems (PAAMS)*, Advances in Soft Computing, pages 440–449, 2009.

[CANB10] N. Criado, E. Argente, P. Noriega, and V. Botti. Towards a Normative BDI Architecture for Norm Compliance. In *Proc. of the International Workshop on Coordination, Organization, Institutions and Norms in Agent Systems (COIN)*, pages 65–81, 2010.

[CANB12a] N. Criado, E. Argente, P. Noriega, and V. Botti. A Distributed Architecture for Enforcing Norms in Open MAS. In *Advanced Agent Technology*, pages 457–471, 2012.

[CANB12b] N. Criado, E. Argente, P. Noriega, and V. Botti. Determining the Willingness to Comply With Norms (Extended Abstract). In *Proc. of the International Conference on Autonomous Agents and Multiagent Systems (AAMAS)*, 2012.

[Cas99] C. Castelfranchi. Prescribed mental attitudes in goal-adoption and norm-adoption. *Journal of Artificial Intelligence and Law*, 7(1):37–50, 1999.

[Cas03] C. Castelfranchi. Formalising the informal?: Dynamic social order, bottom-up social control, and spontaneous normative relations. *Journal of Applied Logic*, 1(1-2):47–92, 2003.

[CC95] R. Conte and C. Castelfranchi. Norms as mental objects. from normative beliefs to normative goals. *From reaction to cognition*, pages 186–196, 1995.

[CCD99] R. Conte, C. Castelfranchi, and F.P.M. Dignum. Autonomous norm acceptance. In *Proc. of the International Workshop on Intelligent Agents, Agent Theories, Architectures, and Languages (ATAL)*, pages 99–112, 1999.

[CDJT00] C. Castelfranchi, F.P.M. Dignum, C. Jonker, and J. Treur. Deliberative normative agents: Principles and architecture. *Intelligent Agents VI. Agent Theories Architectures, and Languages*, pages 364–378, 2000.

[CFV02] M. Colombetti, N. Fornara, and M. Verdicchio. The role of institutions in multiagent systems. In *Proc. of the Workshop on Knowledge based and reasoning agents (AI* IA)*, 2002.

[CGS11] A. Casali, L. Godo, and C. Sierra. A graded bdi agent model to represent and reason about preferences. *Artificial Intelligence*, 175(7-8):1468 – 1478, 2011.

[CHAJ10a] N. Criado, S. Heras, E. Argente, and V. Julian. Contract Argumentation in Virtual Organizations. In *Proc. of the International Workshop on Normative Multiagent Systems (NORMAS)*, pages 55–59, 2010.

[CHAJ10b] N. Criado, S. Heras, E. Argente, and V. Julian. Normative Argumentation. In *Proc. of the International Conference on Practical Applications of Agents and Multi-Agent Systems (PAAMS)*, volume 71, pages 29–36, 2010.

[Che80] B.F. Chellas. *Modal Logic: An Introduction*. Cambridge University Press, Cambridge, England, 1980.

[Chi63] R.M. Chisholm. Contrary-to-duty imperatives and deontic logic. *Analysis*, 24(2):33–36, 1963.

[CJA09] N. Criado, V. Julian, and E. Argente. Towards the implementation of a normative reasoning process. In *Proc. of the International Conference on Practical Applications of Agents and Multi-Agent Systems (PAAMS)*, volume 55, pages 319–328, 2009.

[CJBA09] N. Criado, V. Julian, V. Botti, and E. Argente. A Norm-based Organization Management System. In *Proc. of the International Workshop on Coordination, Organization, Institutions and Norms in Agent Systems (COIN)*, pages 1–16, 2009.

[CJBA10] N. Criado, V. Julian, V. Botti, and E. Argente. A norm-based organization management system. In *Coordination, Organizations, Institutions, and Norms in Agent Systems V*, pages 19–35. Springer, 2010.

[CO07] H.L. Cardoso and E. Oliveira. Institutional reality and norms: Specifying and monitoring agent organizations. *International Journal of Cooperative Information Systems*, 16(1):67–95, 2007.

[CR09] G. Christelis and M. Rovatsos. Automated norm synthesis in an agent-based planning environment. In *Proc. of the International Conference on Autonomous Agents and Multiagent Systems (AAMAS)*, pages 161–168, 2009.

[Cra06] S. Cranefield. A rule language for modelling and monitoring social expectations in multi-agent systems. In *Coordination, Organizations, Institutions, and Norms in Multi-Agent Systems*, pages 246–258. Springer, 2006.

[Cra07] S. Cranefield. Modelling and monitoring social expectations in multi-agent systems. In *Coordination, Organizations, Institutions, and Norms in Agent Systems II*, pages 308–321. Springer, Berlin Heidelberg, 2007.

[Cri11] N. Criado. Reasoning About Norms Within Uncertain Environments (Extended Abstract). In *Proc. of the International Conference on Autonomous Agents and Multiagent Systems (AAMAS)*, pages 1331–1332, 2011.

[CRK90] R.B. Cialdini, R.R. Reno, and C.A. Kallgren. A focus theory of normative conduct: Recycling the concept of norms to reduce littering in public places. *Journal of Personality and Social Psychology*, 58(6):1015–1026, 1990.

[CRP10] G. Christelis, M. Rovatsos, and R.P.A. Petrick. Exploiting Domain Knowledge to Improve Norm Synthesis. In *Proc. of the International Conference on Autonomous Agents and Multiagent Systems (AAMAS)*, pages 831–838, 2010.

[CS06] A.K. Chopra and M.P. Singh. Contextualizing commitment protocol. In *Proc. of the International Conference on Autonomous Agents and Multiagent Systems (AAMAS)*, pages 1345–1352, 2006.

[CVRA09] N. Criado, E. Del Val, M. Rebollo, and E. Argente. A Logic for Normative Time-Bounded Services. In *Conferencia de la Asociación Española para la Inteligencia Artificial (CAEPIA)*, pages 645–654, 2009.

[CW99] R.T. Clemen and R.L. Winkler. Combining probability distributions from experts in risk analysis. *Risk Analysis*, 19(2):187–203, 1999.

[CW07] S. Cranefield and M. Winikoff. Verifying social expectations by model checking truncated paths. In *Proc. of the International Workshop on Coordination, Organization, Institutions and Norms in Agent Systems (COIN)*, 2007.

[DDM02] A. Daskalopulu, T. Dimitrakos, and T. Maibaum. Evidence-based electronic contract performance monitoring. *Group Decision and Negotiation*, 11(6):469–485, 2002.

[DGG95] K. Dittrich, S. Gatziu, and A. Geppert. The active database management system manifesto: A rulebase of ADBMS features. *Rules in Database Systems*, pages 1–17, 1995.

[Dig99] F.P.M. Dignum. Autonomous agents with norms. *Journal of Artificial Intelligence and Law*, 7(1):69–79, 1999.

[DKS02] F.P.M. Dignum, D. Kinny, and L. Sonenberg. From desires, obligations and norms to goals. *Cognitive Science Quarterly*, 2(3-4):407–430, 2002.

[DMSC00] F.P.M. Dignum, D. Morley, EA Sonenberg, and L. Cavedon. Towards socially sophisticated BDI agents. In *Proc. of the International Joint Conference on Autonomous Agents and Multiagent Systems (AAMAS)*, pages 111–118, 2000.

[DMWK96] F.P.M. Dignum, J-J.Ch. Meyer, R. Wieringa, and R. Kuiper. A modal approach to intentions, commitments and obligations: Intention plus commitment yields obligation. In *Proc. of the International Workshop on Deontic Logic in Computer Science (DEON)*, pages 80–97, 1996.

[DP85] D. Dubois and H. Prade. A review of fuzzy set aggregation connectives. *Information sciences*, 36(1-2):85–121, 1985.

[dPSS08] A.P. de Pinninck, C. Sierra, and M. Schorlemmer. Distributed Norm Enforcement via Ostracism. In *Coordination, organizations, institutions, and norms in agent systems III*, pages 301–315. Springer, 2008.

[DR00] E.L. Deci and R.M. Ryan. The" what" and" why" of goal pursuits: Human needs and the self-determination of behavior. *Psychological Inquiry*, 11(4):227–268, 2000.

[dS07] V. Torres da Silva. Implementing norms that govern non-dialogical actions. In *Proc. of the International Workshop on Normative Multiagent Systems (NORMAS)*, 2007.

[dSNdSdL11] B. dos Santos Neto, V. da Silva, and C. de Lucena. Using jason to develop normative agents. *Advances in Artificial Intelligence–SBIA 2010*, pages 143–152, 2011.

[DTM09] M. Dastani, N.A.M. Tinnemeier, and J-J.Ch. Meyer. A programming language for normative multi-agent systems. *Multi-Agent Systems: Semantics and Dynamics of Organizational Models*, pages 397–417, 2009.

[dVCC+10] E. del Val, N. Criado, C. Carrascosa, V. Julian, M. Rebollo, E. Argente, and V. Botti. THOMAS: A Service-Oriented Framework For Virtual Organizations. In *Proc. of the International Conference on Autonomous Agents and Multiagent Systems (AAMAS)*, pages 1631–1632, 2010.

[dVCR+09] E. del Val, N. Criado, M. Rebollo, E. Argente, and V. Julian. Service-Oriented Framework for Virtual Organizations. In *Proc. of the International Conference on Artificial Intelligence (ICAI)*, volume 1, pages 108–114, 2009.

[DVSD04] V. Dignum, J. Vázquez-Salceda, and F.P.M. Dignum. A model of almost everything: Norms, structure and ontologies in agent organizations. In *Proc. of the International Conference on Autonomous Agents and Multiagent Systems (AAMAS)*, volume 3, pages 1498–1499. IEEE Computer Society, 2004.

[DVSD05] V. Dignum, J. Vázquez-Salceda, and F.P.M. Dignum. Omni: Introducing social structure, norms and ontologies into agent organizations. *Programming Multi-Agent Systems*, pages 181–198, 2005.

[EdlCS02] M. Esteva, D. de la Cruz, and C. Sierra. ISLANDER: an electronic institutions editor. In *Proc. of the first international joint conference on Autonomous agents and multiagent systems (AAMAS)*, volume 3, pages 1045–1052, 2002.

[Els89] J. Elster. Social norms and economic theory. *Journal of Economic Perspectives*, 3(4):99–117, 1989.

[Els96] J. Elster. Rationality and the Emotions. *The Economic Journal*, 106(438):1386–1397, 1996.

[Els00] J. Elster. *Strong feelings: Emotion, addiction, and human behavior*. The MIT Press, 2000.

[Eme90] E. Allen Emerson. Temporal and modal logic. In *Handbook of Theoretical Computer Science*, volume B, pages 995–1072. North-Holland, Amsterdam, 1990.

[Eps01] J.M. Epstein. Learning to be thoughtless: Social norms and individual computation. *Computational Economics*, 18(1):9–24, 2001.

[Est02] M. Esteva. *Electronic Institutions: From Especification To Development*. PhD thesis, Universitat Politècnica de Catalunya, 2002.

[Etz64] A. Etzioni. *Modern Organizations (Foundations of Modern Sociology)*. Prentice Hall, 1964.

[Fag03] R. Fagin. *Reasoning about knowledge*. The MIT Press, 2003.

[FAS+10] R.L. Fogués, J. M. Alberola, J. M. Such, A. Espinosa, and A. García-Fornes. Towards Dynamic Agent Interaction Support in Open Multiagent Systems. In *Proc. of the International Conference of the Catalan Association for Artificial Intelligence (CCIA)*, volume 220, pages 89–98. IOS Press, 2010.

[FC02] N. Fornara and M. Colombetti. Operational specification of a commitment-based agent communication language. In *Proc. of the International Conference on Autonomous Agents and Multiagent Systems (AAMAS)*, volume 2, page 542, 2002.

[FC08] N. Fornara and M. Colombetti. Specifying and enforcing norms in artificial institutions. In *Proc. of the International Conference on Autonomous Agents and Multiagent Systems (AAMAS)*, volume 3, pages 1481–1484, 2008.

[FH71] D. Føllesdal and R. Hilpinen. Deontic logic: An introduction. *Deontic logic: Introductory and systematic readings*, pages 1–35, 1971.

[FKT01] I. Foster, C. Kesselman, and S. Tuecke. The anatomy of the grid: Enabling scalable virtual organizations. *International Journal of High Performance Computing Applications*, 15(3):200, 2001.

[FMB05] J. Ferber, F. Michel, and J. Baez. Agre: Integrating environments with organizations. *Environments for Multi-agent Systems*, pages 48–56, 2005.

[For84] J.W. Forrester. Gentle murder, or the adverbial Samaritan. *The Journal of Philosophy*, 81(4):193–197, 1984.

[FPU01] F. Flentge, D. Polani, and T. Uthmann. Modelling the emergence of possession norms using memes. *Journal of Artificial Societies and Social Simulation*, 4(4), 2001.

[FVC07] N. Fornara, F. Viganò, and M. Colombetti. Agent communication and artificial institutions. *Autonomous Agents and Multi-Agent Systems*, 14(2):121–142, 2007.

[FvSM06] J. Fix, C. von Scheve, and D. Moldt. Emotion-based norm enforcement and maintenance in multi-agent systems: foundations and petri net modeling. In *Proc. of the International Conference on Autonomous Agents and Multiagent Systems (AAMAS)*, pages 105 – 107, 2006.

[GAD07] D. Grossi, H. Aldewereld, and F.P.M. Dignum. Ubi lex, ibi poena: Designing norm enforcement in e-institutions. In *Coordination, Organizations, Institutions, and Norms in Agent Systems II*, pages 101–114. Springer, 2007.

[Gae08] D. Gaertner. *Argumentation and Normative Reasoning*. PhD thesis, University of London, 2008.

[GAVSD06] D. Grossi, H. Aldewereld, J. Vázquez-Salceda, and F.P.M. Dignum. Ontological aspects of the implementation of norms in agent-based electronic institutions. *Computational & Mathematical Organization Theory*, 12(2):251–275, 2006.

[GC07] A. García-Camino. Ignoring, forcing and expecting concurrent events in electronic institutions. In *Coordination, Organization, Institutions and Norms in Agent Systems III*, pages 15–26. Springer, 2007.

[GCNRA05] A. García-Camino, P. Noriega, and J.A. Rodríguez-Aguilar. Implementing norms in electronic institutions. In *Proc. of the European Workshop on Multi-agent Systems (EUMAS)*, pages 482–483, 2005.

[GCRASV06] A. García-Camino, J.A. Rodríguez-Aguilar, C. Sierra, and W.W. Vasconcelos. Norm-oriented programming of electronic institutions. In *Proc. International Conference on Autonomous Agents and Multiagent Systems (AAMAS)*, pages 670–672, 2006.

[GD05] D. Grossi and F.P.M. Dignum. From abstract to concrete norms in agent institutions. *Formal Approaches to Agent-Based Systems*, pages 12–29, 2005.

[GGCN+07] D. Gaertner, A. Garcia-Camino, P. Noriega, J.A. Rodriguez-Aguilar, and W. Vasconcelos. Distributed norm management in regulated multiagent systems. In *Proc. of the International Conference on Autonomous Agents and Multiagent Systems (AAMAS)*, pages 624–631, 2007.

[Gib65] J.P. Gibbs. Norms: The problem of definition and classification. *The American Journal of Sociology*, 70:586–594, 1965.

[Giu93] F. Giunchiglia. Contextual reasoning. *Epistemologia, special issue on I Linguaggi e le Macchine*, 16:345–364, 1993.

[GMD06] D. Grossi, J-J.Ch. Meyer, and F.P.M. Dignum. Counts-as: Classification or constitution? An answer using modal logic. *Deontic Logic and Artificial Normative Systems*, pages 115–130, 2006.

[GP04] M.A. Garcia and D. Puig. Robust aggregation of expert opinions based on conflict analysis and resolution. *Current Topics in Artificial Intelligence*, pages 488–497, 2004.

[GR04] G. Governatori and A. Rotolo. Defeasible logic: Agency, intention and obligation. *Deontic Logic*, pages 114–128, 2004.

[GR08] G. Governatori and A. Rotolo. Changing legal systems: Abrogation and annulment part i: Revision of defeasible theories. In *Proc. of the Conference on Deontic Logic in Computer Science*, pages 3–18, 2008.

[GRS05] G. Governatori, A. Rotolo, and G. Sartor. Temporalised normative positions in defeasible logic. In *Proc. of the international conference on Artificial intelligence and law (ICAIL)*, pages 25–34, 2005.

[GS94] F. Giunchiglia and L. Serafini. Multilanguage hierarchical logics, or: How we can do without modal logics. *Artificial Intelligence*, 65(1):29–70, jan 1994.

[Háj98] P. Hájek. *Metamathematics of fuzzy logic*, volume 4. Springer, 1998.

[Han69] B. Hansson. An analysis of some deontic logics. *Noûs*, 3:373–398, 1969.

[Han90] S.O. Hansson. Preference-based deontic logic (PDL). *Journal of Philosophical Logic*, 19(1):75–93, 1990.

[Han04] J. Hansen. Problems and results for logics about imperatives. *Journal of Applied Logic*, 2(1):39–61, 2004.

[Han09] S.O. Hansson. Logic of belief revision. In *The Stanford Encyclopedia of Philosophy*. Spring 2009 edition, 2009.

[Har84] D. Harel. Dynamic logic. In Dov Gabbay and F. Guenthner, editors, *Handbook of philosophical logic*, chapter II.10, pages 497–604. Reidel, 1984.

[HBKR10a] J.F. Hübner, O. Boissier, R. Kitio, and A. Ricci. Instrumenting multi-agent organisations with organisational artifacts and agents. *Autonomous Agents and Multi-Agent Systems*, 20(3):369–400, 2010.

[HBKR10b] J.F. Hübner, O. Boissier, R. Kitio, and A. Ricci. Instrumenting multi-agent organisations with organisational artifacts and agents. *Autonomous Agents and Multi-Agent Systems*, 20(3):369–400, 2010.

[HCAJ09a] S. Heras, N. Criado, E. Argente, and V. Julian. A Dialogue-Game Approach for Norm-based MAS Coordination. In *International Conference on Hybrid Artificial Intelligence Systems (HAIS)*, pages 468–475, 2009.

[HCAJ09b] S. Heras, N. Criado, E. Argente, and V. Julian. Norm Emergency through Argumentation. *Journal of Physical Agents*, 3:31–38, 2009.

[HPvdT07] J. Hansen, G. Pigozzi, and L. van der Torre. Ten philosophical problems in deontic logic. In *Proc. of the International Workshop on Normative Multiagent Systems (NORMAS)*, 2007.

[JcSSD10] S. Joseph, c. Sierra, M. Schorlemmer, and P. Dellunde. Deductive coherence and norm adoption. *Logic Journal of the IGPL*, page In Press, 2010.

[Jøs01] A. Jøsang. A logic for uncertain probabilities. *International Journal of Uncertainty, Fuzziness and Knowledge-Based Systems*, 9(3):279–311, 2001.

[JS93] A.J.I. Jones and M. Sergot. On the characterisation of law and computer systems: The normative systems perspective. *Deontic logic in computer science: normative system specification*, pages 275–307, 1993.

[JS96] A.J.I. Jones and M. Sergot. A formal characterisation of institutionalised power. *Logic Journal of IGPL*, 4(3):427, 1996.

[Kan71] S. Kanger. New foundations for ethical theory. *Hilpinen*, pages 36–58, 1971.

[Kan72] S. Kanger. Law and logic. *Theoria*, 38:105–132, 1972.

[KMS+04] A. Kakas, P. Mancarella, F. Sadri, K. Stathis, and F. Toni. The KGP model of agency. In *Proc. of the European Conference on Artificial Intelligence (ECAI)*, page 33, 2004.

[KN03] M.J. Kollingbaum and T.J. Norman. NoA-a normative agent architecture. In *Proc. of the International Joint Conference on Artificial intelligence (IJCAI)*, volume 18, pages 1465–1466, 2003.

[KNPS06] M.J. Kollingbaum, T.J. Norman, A. Preece, and D. Sleeman. Norm refinement - informing the re-negotiation of contracts. In *Proc. of the International Workshop on Coordination, Organization, Institutions and Norms in Agent Systems (COIN)*, pages 46–51, 2006.

[Kol05] M.J. Kollingbaum. *Norm-governed practical reasoning agents*. PhD thesis, University of Aberdeen, 2005.

[KPT02] G.A. Kaminka, D.V. Pynadath, and M. Tambe. Monitoring teams by overhearing: A multi-agent plan-recognition approach. *Journal of Artificial Intelligence Research*, 17(1):83–135, 2002.

[Kra75] E.F. Krause. *Taxicab geometry*. Addison Wesley Publishing Company, 1975.

[Kri63] S. Kripke. Semantical considerations on modal logic. *Acta Philosophica Fennica*, 16:83–94, 1963.

[Kro87] F. Kroger. *Temporal Logic of Programs*. Springer-Verlag, 1987.

[Lew74] D.K. Lewis. Semantic analyses for dyadic deontic logic. *Logical theory and semantic analysis: Essays dedicated to Stig Kanger on his fiftieth birthday*, pages 1–14, 1974.

[Lin77] L. Lindahl. *Position and Change: A Study in Law and Logic*. Springer, 1977.

[LM08] M. Luck and P. McBurney. Computing as interaction: Agent and agreement technologies. In *Proc. of the European Workshop on Multi-agent Systems (EUMAS)*, pages 1–15, 2008.

[LMSW05] M. Luck, P. McBurney, O. Shehory, and S. Willmott. Agent technology: Computing as interaction: A roadmap for agent-based computing. Technical report, Agentlink, 2005.

[LT03] F. Legras and C. Tessier. Lotto: Group formation by overhearing in large teams. In *Proc. of the International Conference on Autonomous Agents and Multiagent Systems (AAMAS)*, pages 425–432, 2003.

[LyLL02] F. López y López and M. Luck. A model of normative multi-agent systems and dynamic relationships. In *Proc. of the Workshop on Regulated Agent-Based Social Systems*, pages 259–280, 2002.

[LyLL03] F. López y López and M. Luck. Modelling norms for autonomous agents. In *Proc. of the Mexican International Conference on Computer Science (ENC)*, pages 238–245, 2003.

[LyLLd02] F. López y López, M. Luck, and M. d'Inverno. Constraining autonomy through norms. In *Proc. of the International Conference on Autonomous Agents and Multiagent Systems (AAMAS)*, pages 674–681, 2002.

[LyLLd06] F. López y López, M. Luck, and M. dInverno. A normative framework for agent-based systems. *Computational & Mathematical Organization Theory*, 12(2):227–250, 2006.

[Mal71] E. Mally. Logische schriften. groes logikfragment grundgesetze des sollens. *Dordrecht: Reidel*, pages 227–324, 1971.

[McN10] P. McNamara. Deontic logic. In *The Stanford Encyclopedia of Philosophy*. Summer 2010 edition, 2010.

[Mén00] C. Ménard. Enforcement procedures and governance structures: what relationship. *Institutions, Contracts and Organizations, Edward Elgar*, pages 234–253, 2000.

[Mey87] J.J.C. Meyer. A different approach to deontic logic: deontic logic viewed as a variant of dynamic logic., 1987.

[MFM+09] S. Modgil, N. Faci, F. Meneguzzi, N. Oren, S. Miles, and M. Luck. A framework for monitoring agent-based normative systems. In *Proc. of the International Joint Conference on Autonomous Agents and Multiagent Systems (AAMAS)*, pages 153–160, 2009.

[MFPF01] M. T. Martinez, P. Fouletier, K. H. Park, and J. Favrel. Virtual enterprise - organisation, evolution and control. *International Journal of Production Economics*, 74(1-3):225–238, 2001.

[MMH02] L. Mui, M. Mohtashemi, and A. Halberstadt. Notions of reputation in multi-agents systems: a review. In *Proc. of the International Conference on Autonomous Agents and Multiagent Systems (AAMAS)*, volume 1, pages 280–287, 2002.

[Mor56] R. T. Morris. A typology of norms. *American Sociological Review*, 31:610–613, 1956.

[MT95] Y. Moses and M. Tennenholtz. Artificial social systems. *Computers and Artificial Intelligence*, 14(6), 1995.

[MU98] N.H. Minsky and V. Ungureanu. A mechanism for establishing policies for electronic commerce. In *International Conference on Distributed Computing Systems*, pages 322–331, 1998.

[MU00] N.H. Minsky and V. Ungureanu. Law-governed interaction: a coordination and control mechanism for heterogeneous distributed systems. *ACM Transactions on Software Engineering and Methodology (TOSEM)*, 9(3):273–305, 2000.

[MvdT00] D. Makinson and L. van der Torre. Input/output logics. *Journal of Philosophical Logic*, 29(4):383–408, 2000.

[MvdT03] D. Makinson and L. van der Torre. Permission from an input/output perspective. *Journal of Philosophical Logic*, 32(4):391–416, 2003.

[MWD98] J-J.Ch. Meyer, R.J. Wieringa, and FPM Dighum. The role of deontic logic in the specification of information systems. *Kluwer Academic Publishers*, pages 71–116, 1998.

[NL04] M. Nakamaru and S.A. Levin. Spread of two linked social norms on complex interaction networks. *Journal of theoretical biology*, 230(1):57–64, 2004.

[NS05] T. Nakano and T. Suda. Self-organizing network services with evolutionary adaptation. *IEEE Transactions on Neural Networks*, 16(5):1269–1278, 2005.

[Nut97] D. Nute. *Defeasible deontic logic*. Kluwer Academic Pub, 1997.

[Nut03] D. Nute. Defeasible logic. *Web Knowledge Management and Decision Support*, pages 151–169, 2003.

[OCC88] A. Ortony, G. Clore, and A. Collins. *The Cognitive Structure of Emotions*. Cambridge University Press, 1988.

[OLMN08] N. Oren, M. Luck, S. Miles, and T.J. Norman. An argumentation inspired heuristic for resolving normative conflict. In *Proc. of the International Workshop on Coordination, Organization, Institutions and Norms in Agent Systems (COIN)*, pages 1–15, 2008.

[OPVS+09] N. Oren, S. Panagiotidi, J. Vázquez-Salceda, S. Modgil, M. Luck, and S. Miles. Towards a formalisation of electronic contracting environments. *Coordination, Organizations, Institutions and Norms in Agent Systems IV*, pages 156–171, 2009.

[ORV08] A. Omicini, A. Ricci, and M. Viroli. Artifacts in the A&A meta-model for multi-agent systems. *Autonomous Agents and Multi-Agent Systems*, 17(3):432–456, 2008.

[Pav79] J. Pavelka. On fuzzy logic i, ii, iii. *Z. Math. Logik Grundlagen Math*, 25:4552, 119–134, 447–464, 1979.

[Plo81] G. D. Plotkin. *A Structural Approach to Operational Semantics*. University of Aarhus, University of Aarhus, 1981.

[Pos96] E. Posner. The regulation of solidary groups: The influence of legal and nonlegal sanctions on collective action. *University of Chicago Law Review*, 63:97–133, 1996.

[PSJ98] S. Parsons, C. Sierra, and N.R. Jennings. Agents that reason and negotiate by arguing. *Journal of Logic and Computation*, 8(3):261–292, 1998.

[PSMDP12] Isaac Pinyol, Jordi Sabater-Mir, Pilar Dellunde, and Mario Paolucci. Reputation-based decisions for logic-based cognitive agents. *Autonomous Agents and Multi-Agent Systems*, 24:175–216, 2012.

[Puj06] J.M. Pujol. *Structure in artificial societies*. PhD thesis, Universitat Politècnica de Catalunya, 2006.

[RAL03] E.L. Rissland, K.D. Ashley, and RP Loui. AI and law: a fruitful synergy. *Artificial Intelligence*, 150(1-2):15, 2003.

[Rao96] A. Rao. AgentSpeak (L): BDI agents speak out in a logical computable language. *Agents Breaking Away*, pages 42–55, 1996.

[Raw55] J. Rawls. Two concepts of rules. *The Philosophical Review*, 64(1):3–32, 1955.

[RL07] F. Raimondi and A. Lomuscio. Automatic verification of multi-agent systems by model checking via ordered binary decision diagrams. *Journal of Applied Logic*, 5(2):235–251, 2007.

[Ros44] A. Ross. Imperatives and logic. *Philosophy of Science*, 11(1):30–46, 1944.

[RS08] R. Rubino and G. Sartor. Preface. *Journal of Artificial Intelligence and Law*, 16(1):1–5, 2008.

[SA07] S. Sen and S. Airiau. Emergence of norms through social learning. In *Proc. of the International Joint Conference on Artificial Intelligence (IJCAI)*, pages 1507–1512, 2007.

[SB75] E.H. Shortliffe and B.G. Buchanan. A model of inexact reasoning in medicine. *Mathematical Biosciences*, 23(3-4):351–379, 1975.

[SC06] M.J. Sergot and R. Craven. The deontic component of action language nC+. In *Proc. of the International Workshop on Deontic Logic in Computer Science (DEON)*, pages 222–237, 2006.

[SC09] B.T.R. Savarimuthu and S. Cranefield. A categorization of simulation works on norms. In *Proc. of the International Workshop on Normative Multiagent Systems (NORMAS)*, pages 39–58, 2009.

[Sco95] W.R. Scott. *Institutions and organizations*. SAGE Thousand Oaks, 1995.

[Sco02] W.R. Scott. *Organizations: Rational, Natural and Open Systems*. Prentice Hall, 2002.

[SCPP07] B.T.R. Savarimuthu, S. Cranefield, M. Purvis, and M. Purvis. Role model based mechanism for norm emergence in artificial agent societies. In *Proc. of the International Workshop on Coordination, Organization, Institutions and Norms in Agent Systems (COIN)*, pages 1–12, may 2007.

[SDM07] B.R. Steunebrink, M. Dastani, and J-J.Ch. Meyer. A logic of emotions for intelligent agents. In *Proc. of the National Conference on Artificial Intelligence (AAAI)*, pages 142–147. AAAI Press, 2007.

[Sea69] J.R. Searle. *Speech Acts: An Essay in the Philosophy of Language*. Cambridge University Press, 1969.

[Sea97] J.R. Searle. *The Construction of Social Reality*. Free Press, 1997.

[Sea05] J.R. Searle. What is an institution? *Journal of Institutional Economics*, 1(1):1–22, 2005.

[SEGFB11] J. M. Such, A. Espinosa, A. García-Fornes, and V. Botti. Partial Identities as a Foundation for Trust and Reputation. *Engineering Applications of Artificial Intelligence*, 24(7):1128–1136, 2011.

[Ser98] M.J. Sergot. Normative positions. In *Norms, Logics and Information Systems. New Studies in Deontic Logic and Computer Science*, pages 289–310. IOS Press, 1998.

[Ser01] M. Sergot. A computational theory of normative positions. *ACM Transactions on Computational Logic (TOCL)*, 2(4):622, 2001.

[Sha09] S. Shapiro. Classical logic. In Edward N. Zalta, editor, *The Stanford Encyclopedia of Philosophy*. Winter 2009 edition, 2009.

[Sin99] M.P. Singh. An ontology for commitments in multiagent systems. *Journal of Artificial Intelligence and Law*, 7(1):97–113, 1999.

[Sin00] M. Singh. A social semantics for agent communication languages. *Issues in agent communication*, pages 31–45, 2000.

[SPP08] B.T.R. Savarimuthu, Maryam Purvis, and Martin K. Purvis. Social norm emergence in virtual agent societies. In *Proc. of the International Conference on Autonomous Agents and Multiagent Systems (AAMAS)*, pages 1521–1524, 2008.

[SS05] J. Sabater and C. Sierra. Review on computational trust and reputation models. *Artificial Intelligence Review*, 24(1):33–60, 2005.

[SS06] C.S. Sripada and S. Stich. A framework for the psychology of norms. *The Innate Mind: Culture and Cognition*, pages 280–301, 2006.

[SST06] F. Sadri, K. Stathis, and F. Toni. Normative KGP agents. *Computational & Mathematical Organization Theory*, 12(2):101–126, 2006.

[ST92a] Y. Shoham and M. Tennenholtz. Emergent Conventions in Multi-Agent Systems: initial experimental results and observations. In *Proc. of the International Conference on Principles of Knowledge Representation and Reasoning (KR)*, pages 507–521, 1992.

[ST92b] Y. Shoham and M. Tennenholtz. On the synthesis of useful social laws for artificial agent societies (preliminary report). In *Proc. of the National Conference on Artificial Intelligence (AAAI)*, pages 276–276, 1992.

[ST97] Y. Shoham and M. Tennenholtz. On the emergence of social conventions: modeling, analysis, and simulations. *Artificial Intelligence*, 94(1):139–166, 1997.

[Sto61] M. Stone. The linear opinion pool. *Ann. Math. Statist*, 32:1339–1342, 1961.

[TDM10] N.A.M. Tinnemeier, M. Dastani, and J-J.Ch. Meyer. Programming norm change. In *Proc. of the International Conference on Autonomous Agents and Multiagent Systems (AAMAS)*, pages 957–964, 2010.

[Tha00] P. Thagard. *Coherence in Thought and Action*. The MIT Press, Cambridge, Massachusetts, 2000.

[The02] G. Therborn. Back to norms! on the scope and dynamics of norms and normative action. *Current Sociology*, 50:863–880, 2002.

[TKS99] S. Theodoridis, K. Koutroumbas, and R. Smith. *Pattern recognition*. Academic Press, 1999.

[Tuo95] R. Tuomela. *The Importance of Us*. Stanford University Press, 1995.

[TZ00] Z. Tao and T. Zhu. Agency and Self-Enforcing Contracts. *Journal of Comparative Economics*, 28(1):80–94, 2000.

[UBJ+03] A. Uszok, J. Bradshaw, R. Jeffers, N. Suri, P. Hayes, M. Breedy, L. Bunch, M. Johnson, S. Kulkarni, and J. Lott. Kaos policy and domain services: Toward a description-logic approach to policy representation, deconfliction, and enforcement. In *Proc. of Policy*, pages 93–96, 2003.

[UBL+08] A. Uszok, J.M. Bradshaw, J. Lott, M.R. Breedy, L. Bunch, P.J. Feltovich, M. Johnson, and H. Jung. New developments in ontology-based policy management: Increasing the practicality and comprehensiveness of KAoS. In *Proc. of Policy*, pages 145–152, 2008.

[Vas04] W.W. Vasconcelos. Norm verification and analysis of electronic institutions. In *Proc. of the Workshop on Declarative Agent Languages and Technologies (DALT)*, pages 166–182. Springer, 2004.

[VCRA09] E. Del Val, N. Criado, M. Rebollo, and E. Argente. Normative Time-Bounded Service Logic. In *Proc. of the European Workshop on Multi-agent Systems (EUMAS)*, pages 1–13, 2009.

[vdHRW07] W. van der Hoek, M. Roberts, and M. Wooldridge. Social laws in alternating time: Effectiveness, feasibility, and synthesis. *Synthese*, 156(1):1–19, 2007.

[vdTT97] L. van der Torre and Y.H. Tan. The many faces of defeasibility in defeasible deontic logic. *Defeasible Deontic Logic*, pages 79–121, 1997.

[vdTT99a] L. van der Torre and Y.H. Tan. Diagnosis and decision making in normative reasoning. *Journal of Artificial Intelligence and Law*, 7(1):51–67, 1999.

[vdTT99b] L. van der Torre and Y.H. Tan. Rights, duties and commitments between agents. In *Proc. of the International Joint Conference on Artificial Intelligence (IJCAI)*, pages 1239–1246, 1999.

[vdTT01] L. van der Torre and YH Tan. Dynamic normative reasoning under uncertainty: How to distinguish between obligations under uncertainty and prima facie obligations. *Handbook of Defeasible Reasoning and Uncertainty Management Systems*, 6:267–297, 2001.

[Ver00] H. Verhagen. *Norm Autonomous Agents*. PhD thesis, Stockholm University, 2000.

[VKN07] W.W. Vasconcelos, M.J. Kollingbaum, and T.J. Norman. Resolving conflict and inconsistency in norm-regulated virtual organizations. In *Proc. of the International Conference on Autonomous Agents and MultiAgent Systems (AAMAS)*, pages 14–18, 2007.

[Vos01] T. Voss. Game-theoretical perspectives on the emergence of social norms. *Social norms*, pages 105–136, 2001.

[VS99] M. Venkatraman and M.P. Singh. Verifying compliance with commitment protocols. *Autonomous Agents and Multi-Agent Systems*, 2(3):217–236, 1999.

[VS03] J. Vázquez-Salceda. The role of Norms and Electronic Institutions in Multi-Agent Systems applied to complex domains. The HARMONIA framework. *AI Communications*, 16(3):209–212, 2003.

[VSAD04] J. Vázquez-Salceda, H. Aldewereld, and F.P.M. Dignum. Implementing norms in multiagent systems. In *Proc. of the German Conference on Multiagent System Technologies (MATES)*, Lecture Notes in Computer Science, pages 313–327, 2004.

[vW57] G.H. von Wright. Deontic logic. *Logical Studies*, pages 58–74, 1957.

[vW63] G.H. von Wright. *Norm and action: a logical enquiry.* Routledge & Kegan Paul, 1963.

[WJ95] M. Wooldridge and N.R. Jennings. Intelligent agents: Theory and practice. *Knowledge engineering review*, 10(2):115–152, 1995.

[Woo02] M.J. Wooldridge. *An introduction to multiagent systems.* Wiley, 2002.

[WW95] A. Walker and M. Wooldridge. Understanding the emergence of conventions in multi-agent systems. In *Proc. of the International Conference on Multiagent Systems (ICMAS)*, pages 384–390, San Francisco, CA, jun 1995.

[XH09] E. Xiao and D. Houser. Avoiding the sharp tongue: Anticipated written messages promote fair economic exchange. *Journal of Economic Psychology*, 30(3):393–404, 2009.

[YS02] P. Yolum and M.P. Singh. Flexible protocol specification and execution: applying event calculus planning using commitments. In *Proc. of the International Conference on Autonomous Agents and Multiagent Systems (AAMAS)*, pages 527–534, 2002.