UNIVERSITY OF SOUTHAMPTON

Social Power and Norms:

Impact on Agent Behaviour

by

Fabiola López y López

A thesis submitted in partial fulfillment for the
degree of Doctor of Philosophy

in the
Faculty of Engineering and Applied Science
Department of Electronics and Computer Science

June 2003

UNIVERSITY OF SOUTHAMPTON

ABSTRACT

FACULTY OF ENGINEERING AND APPLIED SCIENCE

DEPARTMENT OF ELECTRONICS AND COMPUTER SCIENCE

Doctor of Philosophy

SOCIAL POWER AND NORMS: IMPACT ON AGENT BEHAVIOUR

by Fabiola López y López

Since the agent paradigm emerged, agent researchers have faced the challenge of building *open societies* in which heterogeneous and independently designed entities can work towards similar or different ends. Open societies involve agents that do not necessarily share the same interests, that do not know and might not trust each other, but that can work together and help each other. One of the key omissions in the computational representation of open societies relates to the need for *norms* in multi-agent systems, that help to cope with the heterogeneity, the autonomy and the diversity of interests among their members. This also requires agents that can *reason* about norms because their participation in a society, rather than predefined, must be *voluntary*. So, these agents must understand *why* norms should be adopted and complied with, and *why* the authority and the *power* of agents in a society must be respected. This thesis addresses both the introduction of norms in systems of autonomous agents, and the modelling of agents that can reason about norms.

The thesis makes three main contributions. First, it develops a framework of normative concepts that enables agents to reason about norms and the society in which they participate. Second, it provides the means for agents to identify situations of power, and to use these powers both for the satisfaction of their goals and to understand why the goals of other agents must be satisfied. This is required since agents in an open society must interact with other agents which are also autonomous, and *power* represents a means to influence them. Third, this thesis provides models for agents that adopt and comply with norms not as an end, but as the result of a deliberation process in which their goals and motivations are taken into account. This enables agents to voluntarily decide whether participating in a society is important for the achievement of their goals.

Contents

List of Figures

List of Tables

Acknowledgements

I would like to thank Michael Luck whose excellent supervision, encouragement and advice made me understand the meaning of research. From him I learnt to do research on agents and, above all, I learnt the importance of looking after people I work with. I know that sometimes he helped me beyond his responsibilities as a mentor, and anything I say now will not be enough to express all my gratitude to him. I also thank Mark d'Inverno for his invaluable advices to formalise the theory developed in this thesis. Thanks to Nick Jennings, Cristiano Castelfranchi, Rosaria Conte and all the anonymous reviewers of the papers in which partial results of this work were reported. Their invaluable comments always helped me to improve my work.

I thank the Faculty Enhancement Program (PROMEP) of the Mexican Ministry of Public Education (SEP) and the Benemérita Universidad Autónoma de Puebla (México) for the sponsorship and for the opportunity they gave me to continue my academic career.

Thanks to all the new friends I made in England. They gave me only good moments and made my life, far away from home, easy. To my friends at Warwick: Luz, Maria Stella, Ana, Jose, Carlos, Edgardo, Rebeca, and Carelia. To my friends at Southampton: Sanghee, Angie, Pat, Zhuoan, Jing, Ron, Katia, Steve, Mauricio, Fabiana, Juan Carlos, Arturo, Jorge, Valesca and, above all Cora and Alejandro.

To my always supportive friends in Mexico thanks for their encouragement and for being there when I need you more. To Ana, Lilia, Lulu, Citlalli, Paco, Luis Alfonso, Armando, Rafael, Gabriel, Miguel, Manuel, Martin, Jaime, Esteban, Enrique, Joaquin, Alejandro, Guillermo, and above all Jaime Díaz.

Thanks to my family for their love, support and encouragement. You all know I love you too so much. To Anita, and my nieces Vicky, Priscila, Mariana, Pablis, Heidicita, Lupita, and Mayis. To my mother Isabel and my brothers and sisters: Fidel, Araceli, Alejandro, Raquel, Mario and Eduardo. To my sisters in-law Lucy, Heidy, Sonia and Nidya, and of course to my very large extended family.

To my father, who always lives in my memory.

Chapter 1

Introduction

1.1 Introduction

Agents are software entities able to act without external intervention. Since the agent paradigm emerged, agent researchers have faced the challenge of building *open societies* in which heterogeneous entities can work towards similar or different ends [14, 89]. The basic idea is to enable different software agents to work together in the same way as humans in order to satisfy objectives beyond the abilities of a single agent. However, contrary to models in which agents are designed beforehand to cooperate with each other [66, 97], open societies involve agents that do not necessarily share the same interests, that do not know and might not trust each other, but that can work together and help each other.

Enabling heterogeneous agents to participate in collective work is not easy. Besides basic technological problems regarding communication languages, interaction protocols, ontologies, design methodologies, and standards for all of these, there are problems associated with both the organisation of the participants and the establishment of *norms* that make possible the interaction between these agents. Norms prescribe how agents *ought* to behave in specific situations, and they can make the performance of a system more effective by constraining the behaviour of its components. Norms are also the means of *empowering* less favoured agents by designating other agents as responsible for the satisfaction of their goals. Moreover, it is only through norms that the power and authority of some agents in a society are recognised.

No matter how well the norms of a system are designed or how much power agents have, norms and powers do not have any impact without agents being able to deal with them. That is, for autonomous agents to effectively work in a society regulated by norms some capabilities are required. Agents must be able to adopt and comply with norms, and they must be able to identify and constrain the scope of the authority of empowered agents. Thus, *norms* and *agent powers* are important aspects that must be considered by agent designers to model agents capable of participating in a society.

Although the problem of modelling agents able to deal with norms can be solved by designing agents always to comply with norms and obey authorities, these characteristics constrain agent capabilities to *voluntarily* act in flexible and dynamic environments such as open societies where other agent members are not known in advance, their behaviour is unpredictable, and where norms frequently change, and the normative behaviour of some agents might affect the goals of others. We argue that to effectively act in a society, autonomous agents must *reason* about norms and the power of agents. They must decide, on the basis of their goals and motivations not only when, why and how a norm must be fulfilled, but also when and why an authority must be recognised and its orders obeyed. These aspects of agents and norms are the concern of this thesis.

1.2 Open Societies

At the most elemental level, an open society is a system of heterogeneous entities with imperfect knowledge which, although pursuing their own objectives, can still cooperate with other entities. Open societies are in constant evolution; they are dynamic and, consequently, their members must be able to respond to rapid change in the environment. Since agents are software entities able to act in flexible and unpredictable environments, to take decisions without external control, and to interact with each other, they have been considered by many as suitable metaphors to model open societies. These kinds of societies are needed to implement any kind of system whose members are independently designed, such as virtual organisations, markets and coalitions. There, agents can represent the interests of different users, that do not have to be compatible and yet can converge on some points, just like in human societies. Due to the autonomy of their members, some problems may arise in open societies[35] and actions are taken to solve them. In particular,

- all agent actions that cause negative effects on other agents are constrained, and
- since social actions are performed because agents know what to expect of one another, it is required that these expectations of behaviour are consistent, and that they hold at least until they are satisfied.

Human societies have addressed these problems through *norms* and, by analogy, norms can also be the solution to problems that arise when more than one selfish and autonomous agent meet and interact in a common environment such as in open societies.

1.3 Agents and Multi-Agent Systems

In the last fifteen years, the agent metaphor has been one of the more pervasive issues in computing systems. Its origins can be traced back to the early years of Distributed Artificial Intelligence when people started to take advantage of advances in network technology to solve problems whose complexity requires the effort of several software components [74, 111, 160, 161]. In such models, the idea is to overcome the knowledge, resources, and processing limitations of single components by making several of them work together and, consequently, satisfy complex goals which might not be solved otherwise.

The agent metaphor allows the development of applications in areas as diverse as manufacturing [139], process control [92, 99], business processes [98], personal assistants [110], and health care [88, 109] among others. Nowadays, thanks to both the Internet and the middleware currently available, agents and multi-agent systems are expanding their domains towards applications such as e-commerce, e-markets, virtual organisations, and coalitions [122]. Since in many of these applications agents act on behalf of humans, it is not unusual to find that agent research combines theories of economics, management, sociology, organisation theory, psychology and philosophy with the most recent advances in computer science. Given this interdisciplinary character of agent research, it is common that people use the term *agent* to refer, without distinction, to both human beings and computer systems. Although we share this vision, and many of the concepts and theories discussed in this work also concern humans, the final objectives are directed to facilitate computational systems.

1.4 Autonomous Agents

As mentioned before, it is assumed that the members of an open society are agents independently designed that represent the interests of different users. They are self-interested and autonomous agents that act without external intervention [177]. Their behaviour is directed towards the satisfaction of some goals [23, 90], but when decisions must be taken, their motivations play a key role [117]. Motivations are mechanisms to express an agent's preferences, and they are taken into consideration not only to decide which goal to achieve first, but also whether a goal is preferred over another.

Although autonomous, agents have limits on their abilities, knowledge, and perception, among other things. Thus, the satisfaction of some goals is only possible with the help of other agents, and complex dependence relationships emerge among agents. Dependence relationships have been recognised as the origins of sociality [21, 61, 62].

That is, one of the advantages for agents in a society is that they can overcome their limited capabilities by using the capabilities of others and, in this way, satisfy goals that might not be achieved otherwise. However, since agents are autonomous, they can decide not to help others even if they have the means to do so. Even when an agent agrees to provide help to another, there is no guarantee that such a promise will be kept later on. In addition, conflicts of interest might cause some agents to hinder or even prevent the satisfaction of the goals of other agents. All of this suggests that autonomy must be regulated so that agents become responsible for their actions [116].

1.5 The Importance of Norms

Several researchers have argued that norms play an important role to avoid many of the problems that occur when autonomous agents with different interests converge [11, 38, 44, 45, 164]. Norms are introduced to avoid and solve conflicts, make agreements, reduce complexity, and in general to achieve a desirable social order. It is not difficult to observe that conflicts of interest might appear in a world in which autonomous agents are entirely unconstrained [148].

Norms are mechanisms to regulate the behaviour of agents, and they represent the means by which agents understand their responsibilities towards other agents. In general, these responsibilities are the result of the relationships in which agents have voluntarily agreed to participate. Norms make agents more susceptible to societal concerns, and they allow coherent group actions without centralised control [41, 127]. Norms also represent an agent's expectations about the behaviour of others, and agents work under the belief that others will behave as norms prescribe but, above all, norms are the means of *empowering* agents by entitling them to require other agents to behave in a particular way.

We claim that the use of norms is a necessity in multi-agent systems in which the members are autonomous, but not self-sufficient and, therefore, cooperation is needed, but it cannot be assured.

1.6 Agents: from Autonomous to Normative

To participate in a society regulated by norms, it is enough that agents are able to represent and fulfill norms, and to recognise the authority of certain agents. However, to *voluntarily* be part of a society or to *voluntarily* leave it, other characteristics of agents are needed. Agents must be able to *reason* about why norms should be accepted and complied with. In addition, they must not only be able to recognise the power of agents in a society, but also the limits of this power in order to avoid abusive situations. Since autonomous agents have their own goals and, sometimes, they act on behalf of others whose goals must be satisfied, when reasoning about norms and powers, agents must focus on the positive or negative effects on those goals. Moreover, since agents might have many goals to satisfy and some of them can be in clear conflict with some norms, their motivations must be considered to take effective decisions.

Autonomy to take decisions regarding norms leads us towards the possibility of norm infringement. This negative social behaviour is often found in humans, but could seem an undesirable characteristic for computational entities. Some people might even argue that one of the purposes of norms is to avoid conflicts among agents and to achieve coordination, so that conflicts of interests would arise immediately if agents were allowed to break rules. We agree with this position but we also argue that although norms avoid conflicts between agents, they are the cause of some of them and, consequently, their infringement is important. Thus, norms might be violated in the following cases:

- when agents face two or more conflicting norms, and
- when agent goals are in conflict with norms.

Conflicting norms can be the result of agents belonging to more than one society, in this case, agents must choose which norms to fulfill. Conflicts between two norms can also be a consequence of a poorly designed system of norms. Here, agents can provide useful information to improve these systems. Now, since norms represent something that *ought* to be done, they can be in conflict with some goals. Some overcome this problem by suggesting that in conflicting situations agents must prefer their social responsibilities over their own goals [49, 55]. By contrast, we argue that depending on the motivations of their goals, agents might prefer to suffer the consequences of their acts through punishments instead of losing the opportunity to satisfy one of their important goals.

Autonomy to decide which norms to fulfill enables an agent either to break its relations with other agents or to leave its society in search of one more compatible with its individual goals. This property becomes indispensable to create open societies, coalitions, and any kind of system in which members are designed independently without knowing in advance in which society they can be.

1.7 Aims and Principles

The agent community has been working on norms mainly by modelling multi-agent systems that include norms with which agents are assumed to comply [6, 51, 55, 69, 144, 153]. Others have worked on the representation of norms in particular domains [100], and on the emergence of norms between a group of agents [87, 171]. There is also work that describes some strategies to avoid deviations from norms [28, 85], work that examines the effects in a society of having agents that fulfill their social responsibilities [91, 96, 103], and work that proposes logics for their formalisation [151, 169, 172]. From the perspective of individual agents, agents that always comply with norms have been already modelled [13, 108, 153].

By contrast, little work has been done to explain why autonomous agents adopt and comply with norms [43] and, consequently, models of autonomous agents that reason about norms are scarce, and current proposals are incomplete [29]. This is in part because there is no canonical representation of norms that includes the necessary elements for agents to take effective decisions regarding norms. Thus, for each type of norm, a different process of reasoning is proposed [12, 54]. In addition, since the study of normative systems is focused on the global effects of agents fulfilling their norms (or not), some elements needed by agents to effectively reason about norms and the powers of agents have been omitted. In some systems [69], although agents are able to recognise the power (and authority) of certain agents, they have not been endowed with the abilities to constrain this power. Without this ability, agents are condemned to obey forever such authorities. This contradicts the notion of autonomous decision-making and makes it difficult to represent agents that voluntarily can enter an open society and can leave it when they decide to do so.

Towards the computational implementation of open societies where independently designed, and autonomous agents meet and interact, two aspects demand immediate attention. The first refers to the introduction of norms in multi-agent systems that help to cope with the heterogeneity, the autonomy and the diversity of interests among their members, and the second is related to the development of agents able to reason about the norms and the powers of agents in a society. Now, considering that current research covers only partial aspects of these concerns, the aims of this thesis are focused on an integral conceptualisation of norms, agents that reason about norms, and all the elements in the environment that might influence the normative behaviour of agents. In particular, this thesis is aimed at the following.

- Providing a general model of norms which besides including all the elements needed by agents to take decisions concerning their normative behaviour, allows the representation of the different types of norms that agents have to deal with.

- Providing a model of multi-agent systems regulated by norms that allows agents to understand and delimit the authority of certain agents. In addition, such a model must enable agents to identify the different relationships that emerge between them, not only as a result of norms, but also as a result of the normative decisions of agents. This model must also consider that agents are autonomous and, therefore, although certain behaviour can be expected, it cannot be ensured.

- Providing the means for agents to use the empowered situations that result from both their capabilities and their inclusion in a society to effectively satisfy their goals.

- Providing a model for agents that describes their normative behaviour. In this model, rather than taking for granted the adoption of norms and their compliance, these must be the result of decision-making processes that consider an agent's goals and motivations. These processes must also be coupled with other decision processes because the goals and intentions of agents might be affected by the normative decisions agents make.

These aims will contribute to a better understanding of norms and the role they play in enabling heterogeneous agents to interact each other, and will facilitate the development of more effective agent-based open systems. In developing our theories, there are principles that guide our research. These are listed as follows.

- Since we recognise that autonomous agents might have many goals, we consider that any decision an autonomous agent takes must be not only by evaluating its effects on a particular goal, but by considering overarching effects on all of them. We also adhere to the principle that motivations enable agents to prefer one goal over another, and that the preferred goal is always the most motivated goal at the time.

- We consider that it is not necessary to start from scratch in every new proposal. Well-founded frameworks of agents can be used to build up new and more complex theories. In this way, researchers can concentrate their efforts on the particular focus of their work without addressing issues of inconsistencies at lower levels.

- The anthropomorphic view of agents has encouraged the introduction, in agent research, of theories from other disciplines. The fast deployment of the agent paradigm is a consequence of previous advances in other areas of knowledge.

However, although we recognise the richness that theories from other disciplines might provide, we have to recognise that a direct translation of them is not always possible because, sometimes, these theories are described by using natural languages which introduce vagueness and ambiguities. We also believe that for a concept to be incorporated into agent research, it must be well defined in order both to provide a common vocabulary that can be used in future agent research and to facilitate their computational implementation.

- We certainly believe that agent theories must be supported by their corresponding formalisations, in order to avoid ambiguities introduced by the use of natural languages and to allow their verification. These theories must allow the development of practical systems and, therefore, their formalisations must be directed to facilitate the development of computational systems.

1.8 Thesis Overview

Our research on agents, multi-agent systems and norms is presented in nine chapters, including this one, organised as follows. Chapter 2 reviews work that is used to support the theories expressed in this thesis. It describes research on autonomy and motivations of agents, agent models and architectures, Social Power Theory, and the different perspectives from which researchers have worked on norms.

Very well known concepts such as beliefs, goals, intentions, plans, and motivations are defined in Chapter 3. Once they are defined, they are used as building blocks to define complex components such as agents and autonomous agents. This chapter also discusses the need of formalisms to describe agent's theories, and it describes the Z language as the selected formalism in this thesis. The formal definitions presented in this chapter, although they might look repetitive at first, underpin the model subsequently developed for norms, normative agents, and normative multi-agent systems.

A *normative framework*, as the fundamental part of our theory, is described in Chapter 4. In it, the role and properties of norms are introduced, and a general model of norms is given. By using this model, further categories of norms are also defined. Actions that are either permitted or forbidden by norms are also specified, and the concepts of *norm instances* and *interlocking norms* are introduced. Moreover, a model of multi-agent systems regulated by norms is proposed, and the dynamics that result from norms and the normative behaviour of agents are explained.

Power is recognised as a means to influence the behaviour of autonomous agents in Chapter 5. There, two kinds of agent powers are discussed, and the means for agents to identify empowered situations are provided. *Circumstantial powers* are a consequence of the current goals and capabilities of agents, and *institutional powers* arise from the norms and the roles agents play in a society. Moreover, since agents recognise institutional powers as long as they want to participate in a society, the reasons for agents to become members and to remain in a society are also provided in this chapter.

In Chapter 6, the way in which powers impact the behaviour of agents is explained. In particular, we discuss how four processes of decision-making can change due to the recognition of powers. In goal delegation, agents use their powers to make other agents to satisfy their goals. Then, the selection of plans that require the cooperation of other agents is discussed in terms of an agent's powers. Powers also impact the process of goal adoption because agents without power might be influenced to adopt the goals of agents with power. Finally, norm adoption is given as a way to make formal the adoption of goals.

The reasons why autonomous agents should comply with norms and the way in which any decision regarding norms affects an agent's goals are given in Chapter 7. There, a model of autonomous norm compliance is proposed. This model is divided into two parts, once to deliberate about a norm and the other to update goals accordingly with normative decisions. Nine different ways to take decisions regarding norms are also provided in the chapter.

Chapter 8 provides an experimental way to observe the effects of norm compliance regarding two issues: an agent's performance, and the effectiveness of norms in a system. Finally, Chapter 9 summarises the model, the main contributions and the limitations of this research, and the potential for further investigations.

Parts of the work in this thesis have been already presented and published in different international conferences and workshops through the papers [112, 113, 114, 115, 116].

Chapter 2

Agent Systems and Norms

2.1 Introduction

In order to provide an effective theory for norms, multi-agent systems regulated by norms, and agents that reason about norms and the society in which they participate, current research on *autonomy and motivations*, *agent architectures*, *social power theory*, and different *perspectives of norms* are reviewed in this chapter.

One of the key concepts for explaining the behaviour of agents and their interactions is that of *autonomy* and, although it is mentioned in some research, its meaning and the meaning of other related concepts are not always shared by researchers. Some agent models consider autonomy only with respect to the means of achieving goals and not with the capability to generate goals. Autonomy is also sometimes confused with *asociality* because it is supposed that autonomous agents prefer their own goals over the goals of other agents. This is not always true, and agents sometimes make agreements, join societies, or even provide help in a benevolent way. To explain these phenomena, the preferences of agents and how these preferences are related to their goals must be understood. Thus, autonomy, motivations, goals and preferences are related concepts whose meaning must be understood. Since researchers have used them in many senses [23, 61, 117, 119, 126, 131, 132, 158, 159], the different views of these concepts are reviewed here in order to make our proposals consistent.

Reviewing current agent architectures is necessary firstly because it allows us to understand how agent behaviour is represented by combining different mental elements with different processes of decision and, secondly, because we are not looking for a new model of agents but for the best way to integrate the notion of norms and powers into currently successful models of agents [15, 80, 141].

There is a great deal of literature concerning the problems of how to enable a set of agents to cooperate relating, for instance, to issues of negotiation [128], distributed planning [65] and organisational structures [20]. The work in this thesis, however, is more concerned with the reasons that cause agents to cooperate with each other and to adopt and comply with the norms of a society. Explaining reasons to cooperate has been the main objective of *Social Power Theory* [21, 24, 26, 27, 31, 36] and, since norms are a formal way to commit someone to cooperate, understanding this theory can offer insight into the aims of this thesis.

This review chapter cannot be complete without including research on norms. However, it is not our intention to provide an exhaustive assessment of all the work on norms but just the better known or the most representative research on the topic. Figure 2.1 shows the different perspectives taken into account to assess the state of the art on this issue.

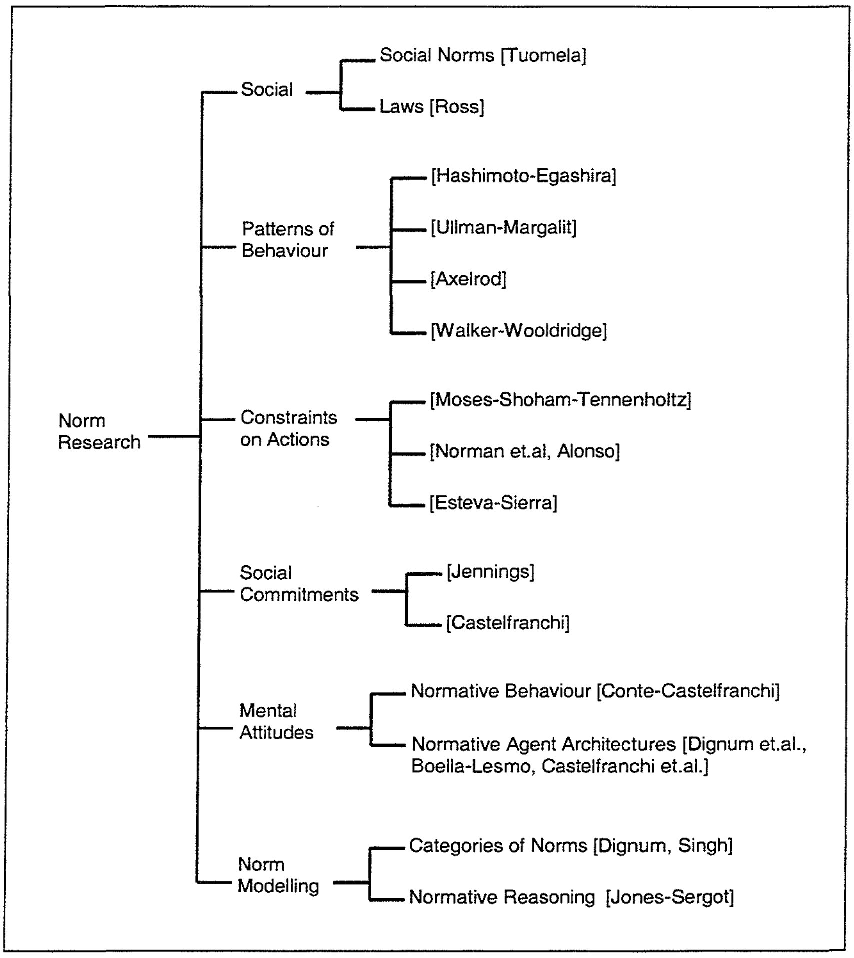


FIGURE 2.1: Research on Norms

Since norms were invented by humans as the means to regulate the behaviour of the members of a society, it is natural to start with the work that describes how norms, their characteristics, and their roles are defined in areas concerned with human beings. In particular, research on norms from the philosophical, social, and legal points of view are described [146, 164, 165, 166]. The maturity already reached in these fields offers us the opportunity to translate many of these theories to the field of agents. We have grouped these approaches as the *social* perspective of norms.

The second perspective considered here aims to explains how some norms emerge in a society of agents (*norms as patterns of behaviour*). In general, approaches of this kind seek to find a pattern of behaviour that works as a norm in a group of agents without involving previous planning [5, 11, 87, 167, 171]. Thus, norms emerge as the result of individuals being forced to make rational choices. Although this perspective does not concern the modelling of agents, it is interesting because it shows how theories already used in other areas such as economics can be expanded to the context of multi-agent systems.

One of the first approaches to introduce norms in agent research is that which describes *norms as constraints on actions*. In this view, norms specify which actions are permitted or forbidden for agents in particular states of a system [17, 127, 153]. Although the original idea has been overtaken, the basic concepts are still used by many [3, 133] and are, therefore, included here.

Social commitments represent agreements to do something between two or more agents. We consider social commitments as norms because they represent the obligation of agents to do something and, in general, social pressure is exerted to make an agent fulfill them. Social commitments have shown their effectiveness to coordinate the activities of agents [174] and, given their importance, we have to review the way researchers have defined and worked with them [22, 93, 94], in order to effectively incorporate the concept of social commitments to our general model of norms.

In the majority of the current research, norms are considered as mental states that might influence agent behaviour [38, 39, 43]. That is, norms are mental attitudes that might produce new goals in an agent and, therefore, they can direct its behaviour. Since a model of agents able to reason about norms is one of the aims of this thesis, approaches of this kind deserve a detailed review [12, 29, 54].

An important trend in norm research has been focussed on defining and specifying the concept of norm, as well as providing classifications for the norms that agents have to deal with [52, 53, 157]. Analysing all this work is necessary in order to find commonalities among the different definitions and *models* of norms. Some research also deals with the problem of modelling reasoning about the norms of a system, and although this is a problem that has emerged in the context of Artificial Intelligence to build expert systems [100, 101], agent researchers use similar mathematics and computational tools [102, 107, 151, 168, 169].

The organisation of this chapter follows the same order in which the different issues were introduced here. Each issue is followed by a brief discussion of the area, and some conclusions drawn at the end.

2.2 Autonomy and Motivations

2.2.1 Introduction

As stated before, related concepts such as autonomy, motivations, preferences and goals have different meanings. Some agent models consider autonomy only with respect to the means of achieving goals. Thus, given a goal, an agent is free to choose the plan which best allows its satisfaction. An agent is also free to change and adapt its plans according to its circumstances, which enables agents to act in dynamic and flexible environments [140]. Autonomy has also been related to the abilities to satisfy a goal without help [23], and the ability to generate goals [61]. Despite these different conceptions, the majority of researchers agree that autonomy is a property that enables agents to take decisions [177] and, to do that, agents consider their preferences or motivations which must be related to their goals. The purpose of this section is to review the different conceptions of autonomy, motivations, preferences, and goals, in order to adopt definitions of them to make our proposals consistent.

2.2.2 Motives and Goals

One of the early efforts to point out the importance of *motivations* was undertaken by Sloman and Croucher [158, 159]. They work on the idea of motives as mechanisms to decide what to do. For these authors, motives represent desires, wishes, tastes, or preferences that can be classified as follows. First-order motives are those which directly specify goals. Second-order motives are subclassified as *motive generators* and *motive comparators* to make a distinction between motives to generate new motives, and motives to give priorities to conflicting motives. Sloman states that the following three parameters must be taken into account for a motive to give rise to a goal: *intensity*, *importance* and *urgency*. Thus, goals with enough intensity are considered by an agent to be intended; however, only the more important goals will be intended. The *urgency* of each goal determines how fast a goal must be satisfied before it is too late.

Moffat and Frijda [126] define *concerns* as dispositions to prefer certain states and/or dislike others. They are related to the fundamental needs of an agent, almost like in the case of a biological organism. Goals are generated every time an event relevant to an agent's concerns is perceived. Then, according to its relevance, the current processing may be interrupted, and a new goal can be intended. Concerns are not active all the time, but are aroused only when relevant events occur in the environment.

2.2.3 Motivated Agency

Contrary to the view that neither goals nor their importance change over time, Norman and Long [131, 132] argue that agents must be able to satisfy more than one goal, and that changes in dynamic environments may lead to changes in goals so that new goals may be created and old goals may be dropped. However, due to the natural limitations of agents, not all the generated goals can be achieved, and agents must limit the number of goals processed at one time by giving them priorities. Norman and Long propose a *motivated agency* architecture based on the BDI model of agents in which motives and motivations play an important role in the generation and selection of goals. They describe a *motive* as a need or desire that causes an agent to act, and a *motivation* as the driving force that arouses and directs actions towards the achievement of goals. In other words, motives are *reasons* for creating goals, and motivation is a measure of the importance of a goal at a particular time. A motivation depends on both the internal state of the agent and the external state of the world. Consequently, goals are seen as the result of different changes that affect the motivations of agents.

In their architecture, the purpose of a motive is to monitor any internal and external changes. Motives are defined as functions that map a list of beliefs to a, possible empty, set of goals. Each goal is associated with a motivation that changes over time and gives relevance to the goal, as well as a criterion to decide which goal to achieve first. Motivation is then defined as a heuristic function which, given a set of beliefs, provides the intensity associated with motives. In this way, a goal is generated only if the intensity of the associated motivation exceeds a predetermined threshold, and only then is the associated motivation mitigated.

2.2.4 Motivation and Autonomy

Although Luck and d'Inverno's view [56, 61, 117, 119] of motivation is also focused on the generation of goals, they go beyond the concept of motivation defined by Norman and Long. They describe motivations as higher-level non-derivative components that provide reasons for doing something. They define *motivation* as any desire or preference that can lead to the generation and adoption of goals, and which affects the outcome of the reasoning or behavioural tasks intended to satisfy these goals. Luck and d'Inverno state that motivation is the main characteristic of autonomous agents because, by having the ability to generate their own goals, they do not depend on the goals of other agents to act.

According to Luck and d'Inverno [119], motivations are associated with goals. Each motivation has a *strength* (or value) that varies over time according to both the internal and external state. This value is used to determine which goal controls the agent behaviour at a particular moment. When this value exceeds a threshold, the agent is said to be motivated to do something and a set of goals is generated [83]. Each autonomous agent can be endowed with a set of motivations whose associated goals depend on the kind of agent being represented. In this way, goals are created and destroyed in order to mitigate an agent's motivations. In Luck and d'Inverno's framework, motivations are also used to solve the problem of conflicting goals. That is, autonomous agents always select a set of goals with the greatest motivation among competing or alternative goals. Goals can also be destroyed when there is not a high enough motivational value that maintains them.

2.2.5 Discussion

In general, most researchers agree that motivations provide reasons to do things, but they differ in the definition of both goals and motivations. Whereas for Sloman motivations are goals, others argue that motivations are not goals because they cannot be represented as states to bring about. We adhere to this latter position, and agree that motivations not only provide reasons to generate goals, but also give reasons to prefer one goal over another, and to hold an intention until the goal becomes achieved [62, 83]. Thus, we adopt d'Inverno and Luck's definition of motivations.

In what follows, *autonomy* is considered as a property that enables agents to act without the intervention of other agents, to have control over their internal state and their behaviour [177]. More specifically, autonomous agents are those that have their own goals, and that are able to take decisions on the basis of their own preferences [23, 117]. We also consider that autonomy must be reflected in the ability of agents not only to choose which goals to pursue, but also to decide which goals to prefer. Having preferences over their goals enables agents to take decisions when conflicts between goals are detected. This acquires special relevance in the cases in which the conflicting goals either belong to other agents or are derived from norms that agents must fulfill. Understanding an agent's preferences allows us to understand those situations in which autonomous agents, although satisfying their own goals, can still provide cooperation. That is, they are able to coexist with other agents.

2.3 Agent Architectures

2.3.1 Introduction

To model the behaviour of agents, four basic approaches are considered: the *reactive*, *deliberative*, *interacting* and *hybrid* models [130]. Reactive agents respond with an immediate action to events that occur in the environment; instances of this model are the *subsumption architecture* [18] and Pengi [1]. Since the main characteristic of reactive agents is that they neither reflect on the long term effects of their actions nor consider the coordination of activities with other agents [177], they are no longer interesting for the purposes of this thesis.

Deliberative agents are those whose behaviour involves different processes of reasoning before making a decision (e.g. BDI architectures [15, 80, 141]), whereas *interacting* agents are those whose architecture includes mechanisms and mental elements to deal with the presence of other agents. For instance, the COSY architecture [19] includes a module for perceiving the external world, and it considers cooperation protocols to allow communication with other agents. Other examples are the agents described in both the GRATE* [97] and ARCHON systems [99]. Finally, *hybrid* architectures are designed to combine the advantages of the different paradigms mentioned above and, typically, they have functional layers to deal with different types of problems. Examples include the InteRRaP [129], Touring Machines [75] and AuRA [4] architectures.

This section reviews key examples of successful architectures by giving particular attention to those that have considered actions that involve other agents. A detailed analysis of all existing architectures is beyond the scope of this thesis, but can be found elsewhere [86, 130, 134, 177].

2.3.2 BDI Architectures

Perhaps one of the best known models of agents is the BDI agent architecture. It is based on the theory of *practical reasoning* stating that an agent's behaviour is driven by its goals. These agents are entirely defined by using three mental attitudes as follows. Beliefs are the representation an agents has about its world, goals are states an agent wants to bring about, and intentions represent the means that an agent has to satisfy its goals. The practical reasoning process is divided into two sub-processes: one for deliberating and deciding what goals an agent wants to achieve (*deliberation*) and the other to decide how those goals should be achieved (*means-ends reasoning*). Once a goal is chosen, it becomes an *intention* that determines the future actions of the agent. Figure 2.2 shows an abstract BDI agent architecture taken from [176]. It shows in ovals

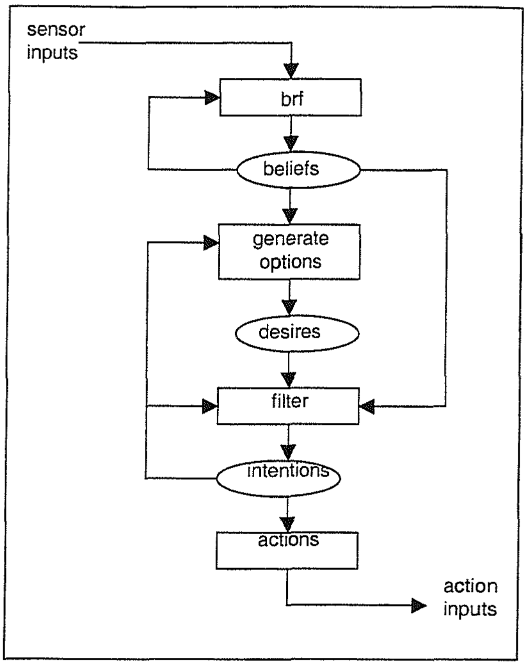


FIGURE 2.2: The BDI Model of Agents

the three main mental attitudes of agents: beliefs, desires and intentions, while boxes represent the decision-making processes as mentioned above.

An early example of a BDI architecture is IRMA (Intelligent Resource-Bounded Machine Architecture) [15], which allows agents to evaluate alternative courses of actions without spending too much effort on the deliberation process. IRMA's main elements are plans, which provide recipes for action to the agent. Besides plans, the model includes beliefs, desires, and a *plan library*, which is a repository of all the plans that one agent knows. Once one of these plans is adopted for execution, it is considered an intention and the agent is committed to it. Now, given that the environment is by default dynamic, agents cannot have complete knowledge of future events and, consequently, they cannot plan in advance all the activities that they have to perform. To solve this problem, instead of having plans that include everything that must be done (*total plans*), incomplete plans are considered (*partial plans*). Partial plans include subgoals to represent desired states, but without a corresponding subplan to achieve them. The selection of this plan is made only at the time at which the subgoal must be satisfied.

IRMA agents have four processes for reasoning: the *means-ends reasoner*, the *opportunity analyser*, the *filtering process* and the *deliberation process*. Figure 2.3 illustrates these processes in rectangles, whereas ovals represent mental states, and arrows indicate the data flow described as follows. For each partial plan already adopted, the *means-ends reasoner* is invoked to propose subplans to complete it. At the same time the

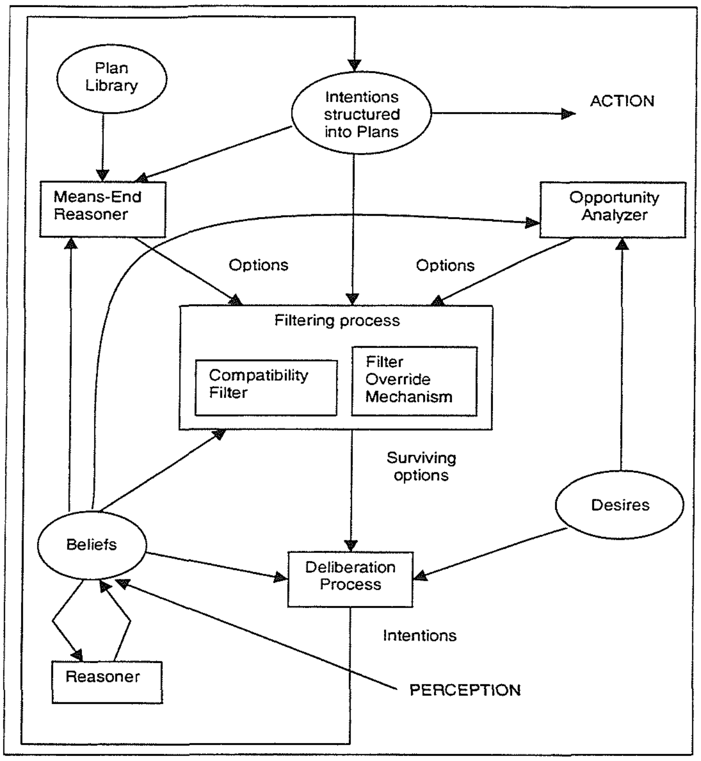


FIGURE 2.3: IRMA Architecture

opportunity analyser also proposes other options (i.e. goals) that result from changes in the beliefs of the agent due to events in the environment. Both proposals are passed to the *compatibility filter* of the *filtering process* in order to check for a conflict with the current intentions. All compatible options are then sent to the *deliberation process*, which is responsible for weighing these competing options against one another in order to produce new intentions that must be incorporated into the existing ones.

Options classified as incompatible by the compatibility filter are sent to the *filter override mechanism* of the *filtering process*. This process is responsible for reconsidering the possibility of dropping some intentions in order to take new opportunities that the environment provides, and then adopting new plans as intentions. In this phase, agents evaluate their options and decide whether to keep the old ones or to change them for new ones. Sometimes, as a result of a change in the beliefs of agents, plans are no longer achievable. In this case, if the plan is a subplan of another, the *means-end reasoner* is invoked again. However, if the plan was adopted to satisfy a desire that still exists, the agent tries to find alternative plans to satisfy it.

Perhaps the best known BDI system is the PRS (Procedural Reasoning System) agent architecture [80], which also instantiates the BDI model and uses partial plans as the means to achieve goals. An abstraction of PRS, and its successor dMARS (distributed

Multi-Agent Reasoning System), can be found in AgentSpeak(L) [141, 142] where its creators develop a formalisation of the operations in such architectures. Further formalisations of them are given in [58] and [59].

2.3.3 GRATE*

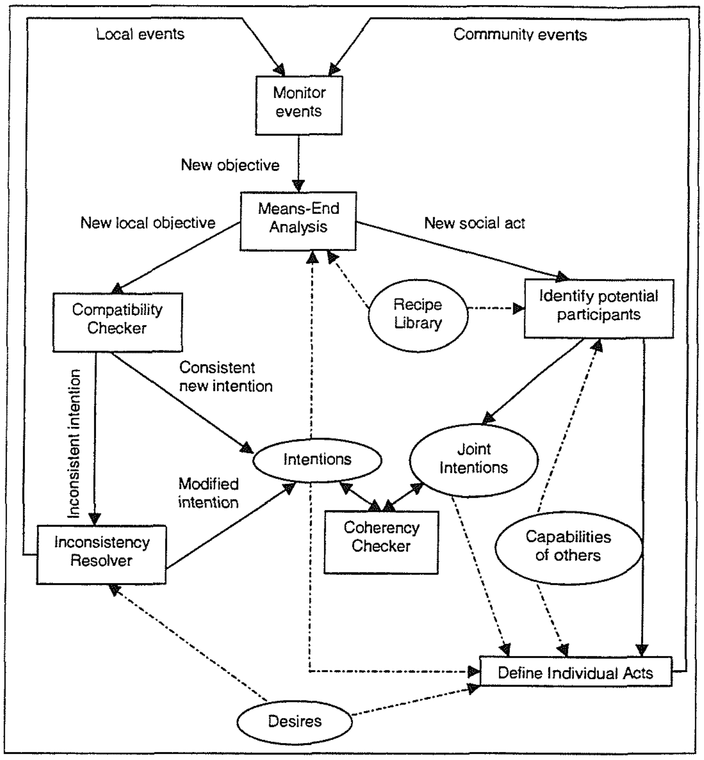


FIGURE 2.4: GRATE Functional Agent Architecture

GRATE* (Generic Rules and Agent Model Testbed) [95, 97] is a general framework to develop multi-agent systems where the individual agent architecture is based on the BDI model, but incorporates capabilities to assess situations in order to determine when a social activity is needed. An agent functional components are shown in Figure 2.4 where ovals are mental states, rectangles are processes, arrows represent the control flow, and dotted arrows the data flow. GRATE agents are able to maintain two roles, one as an individual and the other as a member of a team. In this architecture, goals (identified as tasks) are generated directly from events internally monitored or from the environment. Then, the *means-ends* analyser process decides whether a goal must be achieved locally, or whether it should be delegated to someone else.

If the goal must be locally satisfied, the means-end analyser uses the plan library to find an appropriate plan to fulfill its objective, and creates a *local intention* that is passed either to a *compatibility checker* in order to verify its consistency with other intentions that already exist, or to an *inconsistency resolver* to modify the intention in order to make it consistent with other intentions already adopted. If the goal cannot be performed by the agent, it should be delegated to another agent. Agents use the plan library and information about the capabilities of other agents to determine potential participants in the social action. Once the agent has selected other agents to delegate the goal to, a *joint intention* is created, and its consistency with other local intentions is checked. Finally, both joint and local intentions are executed and monitored until their execution finishes. In addition, agents decide which requests for cooperation should be accepted on the basis of their own capabilities. That is, if the requested goal can be satisfied locally, an agent agrees to cooperate with another.

2.3.4 InteRRaP

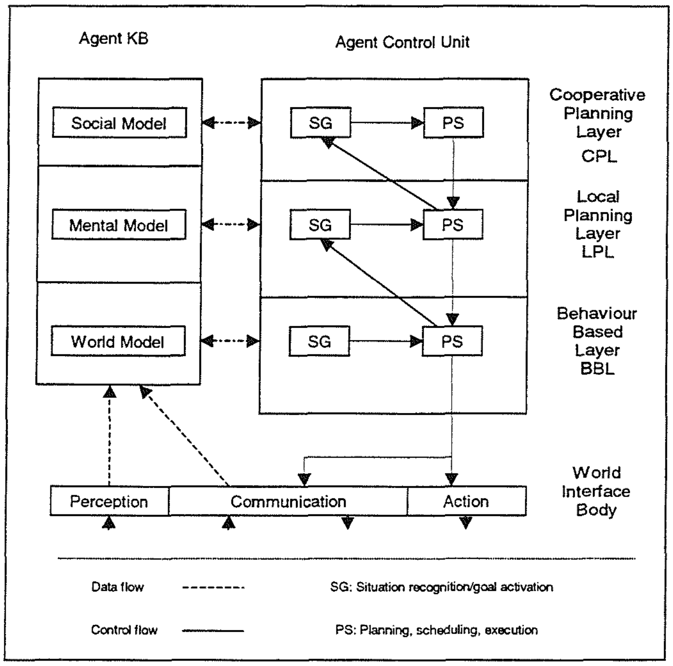


FIGURE 2.5: InteRRaP Architecture

The InteRRaP agent architecture [76, 129] is a hybrid architecture whose structure, shown in Figure 2.5, is divided into three layers: a *reactive*, a *local*, and a *cooperative* layer. The reactive layer is known as the *behaviour-based* layer, and allows an agent to react quickly to changes in the environment. The local layer, called the *plan-based* layer, is similar to the traditional BDI model in that when it is given a goal, a plan is found and then executed. Finally, the third layer, or the *cooperation-based* layer, deals with those goals requiring the cooperation of other agents; it enables agents to interact with other agents by coordinating actions and forming joint plans.

As a consequence of this layered vision, mental states and operational control are also divided into three hierarchical layers. Regarding mental states, beliefs are classified as those that one agent has about its environment (the *world model*), those about the agent itself, including goals, plans and intentions (the *mental model*), and those about other agents, also including joint plans, joint goals and joint intentions (the *social model*). Operation control is similarly divided into three layers in order to deal with three different kinds of goals: reaction goals which immediately activate a process for fast reaction; local goals which are achieved by a local plan; and, finally, those goals that are shared by a group of agents and that need joint plans to be satisfied. Each layer works independently until it recognises a situation it is not able to deal with. In this case, the operation of the next layer in the hierarchy is invoked and control is passed to it.

2.3.5 Discussion

The success of BDI models, such as IRMA and PRS, is a consequence of their flexibility. The model allows attitudes, such as beliefs, desires and intentions, to be explicitly represented and, consequently, easily manipulated and reasoned about. In addition, its control cycle allows agents to detect new events in the environment that can lead to changes in their goals and in their intentions. However, the BDI model does not include any explicit mechanism to interact with other agents, and does not even attempt to explain how and why social interactions between agents occur. Both GRATE* and InteRRaP agents overcome these problems by including mechanisms that consider the existence of other agents and that facilitate the delegation and adoption of goals through the establishment of joint commitments. An interesting point to observe in these models is that both the goal adoption and goal delegation processes are achieved by considering only a few factors. On the one hand, agents decide to delegate goals when these are beyond their capabilities. On the other, agents adopt external goals just in cases in which the satisfaction of these goals is possible according to their capabilities, and are consistent with their current intentions. Thus, there are only two cases in which agents refuse to cooperate: when there is a practical impossibility of achieving the suggested goal, and by having intentions which conflict with that goal. In addition, once an agreement of cooperation is made, agents respect it and the only reason for failing is due to physical failure.

The agent models mentioned above are intended to provide *benevolent* cooperation; the roles for agents are determined in advance, and agreements to cooperate are almost taken for granted. Refusals to cooperate neither affect nor create any kind of relationship between agents. If this happens, it is due to causes beyond an agent's control. The situation in open systems is quite different. Autonomous and self-interested agents coexist and, therefore, cooperation can never be guaranteed. Refusals to cooperate result not only because agents lack capabilities, but also because they can decide not to provide help. Since providing cooperation is an agent's decision, the possibility of changing this decision also exists and, consequently, agreements among agents can also be dropped.

Since the BDI model has been successfully applied in environments where flexibility is needed, and since it has also been successfully augmented with mechanisms to interact and create commitments between agents as in GRATE* and InteRRaP, we believe that it can also be used as the basis for constructing agents with more freedom to choose the goals they want to pursue, the goals they want to adopt, the norms they consider necessary and, above all, the *norms* they want to comply with.

2.4 Social Power Theory

2.4.1 Introduction

Social P.ower Theory states that dependence constitutes the basis of all social interaction, because it is when agents become aware of their dependence on the abilities of other agents to achieve one of their goals that they try to obtain help and, consequently, a process of interaction among agents begins [125]. However, given that other agents are also autonomous and have their own goals, a mechanism to influence them is needed in order to cause them to adopt external goals. In this way, a network of dependence and power among agents is created. Conte and Castelfranchi [38, 40] argue that by making autonomous agents aware of their dependencies, different models of interaction such as cooperation, social exchange, coalitions, negotiation, and even some types of social exploitation, could emerge. All these theories give rise to the *social reasoning mechanism* proposed and simulated by Sichman et al. [46, 154, 155]. This section explains the main concepts underlying this theory.

2.4.2 Social Powers and Dependence

According to Castelfranchi [26], the *personal powers* of agents are determined by their capabilities, resources, skills, knowledge or motivations that allow them to satisfy their goals. When these powers can also be used to satisfy the goals of other agents, relationships of power and dependence are established. Dependence is then considered as a combination of the lack of power of one agent and the corresponding personal powers of another.

Social dependence between two agents occurs when one of them has a goal, and the success of that goal depends on an action which it cannot carry out, while the other agent can [31]. By using this definition as a starting point, more complex dependence relationships between two interacting agents are defined as follows. Firstly, *mutual dependence* is a situation where an agent infers that it, and another agent, socially depend on each other for the same goal; that is, they have a common goal. Secondly, a *reciprocal dependence* situation occurs when both agents socially depend on each other, but for different goals. Finally, *unilateral dependence* is a situation where an agent infers that it socially depends on an agent for one of its goals, but this latter agent does not socially depend on it for any of its goals.

Castelfranchi explains that by using dependence and powers, some strategies to influence agents can be identified. For example, a *promise of a prize* is a strategy where an agent induces another to adopt a goal on the promise of reward (money, welfare, gifts, etc). However, for this strategy to succeed, these prizes must be in accordance with the goals of the agents to be influenced. Castelfranchi also says that a *threat of sanctions* occurs when an agent induces another to adopt one of its goals in order to avoid being punished. In this case, the second agent must know that the first has the power to impose that punishment. By contrast, a *search for cooperation* strategy is used to influence agents to adopt goals because they are pursuing the same goal (mutual dependence). Finally, when two agents are in reciprocal dependence a *future reciprocation* strategy can be used. In this case, one of the agents agrees to adopt a goal on the promise of future help; in other words, an exchange of goals is agreed.

2.4.3 Multiparty Dependence

Social power theory states that an agent not only needs to know its dependence on another specific agent, sometimes it needs to reason about its position within a society. Agents can be in three different situations. First, an agent may find that many agents can help it to overcome its dependence with respect to one of its goals, meaning that it can choose one from among many agents to satisfy its goals. This situation leads the agent to be less socially dependent, because the probability of finding help increases as well as the probability of achieving its goal without having to give something in exchange. On the contrary, if many agents are needed to satisfy its goals, the agent becomes more socially dependent. Finally, when an agent is frequently required for help, so that many agents depend on it, it can be said that it has great *social utility* [40]. These situations are known as *or*, *and* and *co* dependence respectively [31, 40], and a combination of them allows an agent to know its value or importance within a society. This is the *negotiation power* of agents, which determines how useful an agent is for those agents that depend on it.

2.4.4 Discussion

Social Power Theory sets up the basis to explain why many forms of social interactions occur. It contributes to understanding the dynamics of multi-agent systems, and explains how the powers of agents emerge, transform, circulate and multiply when agents are included in a society. In addition, the social reasoning mechanism provides the means for agents to choose a way to interact with other agents. However, this theory is limited to powers that appear due to agent abilities. By contrast, we argue that power is not only given by agent abilities but also by the social structure in which agents exist. Thus, the notion of powers can be extended to include *empowered* situations which are given by the roles agents play in a society. We also consider that powers can be not only used to select strategies for influencing agents, but also as strategies for selecting a plan, which might be the difference between the satisfaction of a goal or not. As can be seen, social power theory still has many things to offer.

2.5 Social Perspective of Norms

2.5.1 Social Norms

From a philosophical point of view, Tuomela proposes a general structure for the norms of a group of agents. Thus, a *social norm* consists of four components: a class of addressee agents, the group of agents to which all addressees belong, the task to be performed by them and, finally, the circumstances under which the task must be carried out [164, 165, 166]. Tuomela classifies social norms as one of two kinds: rules or *r-norms*, and proper social norms or *s-norms*.

According to Tuomela, *rules* represent explicit agreements among agents, and are created by an authority. Rules are subdivided into two further classes as follows. *Formal* rules are those that include legal sanctions such as laws and regulations, and *informal* rules that are not in written form but communicated orally and include informal sanctions.

In addition, *proper social norms* are norms accepted not through agreement but through mutual beliefs, and are also divided into two classes: *conventions*, which concern the whole society or social class and have social sanctions, such as approval or disapproval; and *group-specific norms*, which concern a group of agents in a society.

Tuomela also explains the conditions under which either rules or proper social norms ought to be fulfilled by the members of a group; these conditions cause a norm to be *in force*. Thus, the *promulgation* condition refers to the fact that norms must be issued by an authority. The *accessibility* condition states that all members of the group acquire the belief that they ought to comply with the norm. Now, if many members of the group fulfill the norm, or at least are disposed to do so, it is said that the *pervasiveness* condition is satisfied, whereas the *motivational* condition is met when at least some members sometimes fulfill the norm because they believe it is true and that they ought to do so. The *sanction* condition refers to the existence of social pressure against members that deviate from the norm. Finally, for a rule, the *acceptance* condition is the conjunction of the promulgation and accessibility conditions, whereas for a proper social norm to be accepted, only the accessibility condition is needed. Thus, contrary to rules, s-norms do not need to be issued by an authority, but they have to be recognised as norms for all the members in a group.

Tuomela argues that an *r-norm* is a social *ought-to-do rule* in force in a group if and only if the acceptance (interpreted as promulgation and accessibility), pervasiveness, motivational, and sanction conditions are satisfied. In addition, an s-norm is a *proper social ought-to-do norm* in force in a group if and only if the acceptance (or accessibility), pervasiveness, motivational, and sanction conditions are satisfied.

Besides r-norms and s-norms, Tuomela recognises the existence of other kinds of norms that are not based on social responsiveness, but represent something more personal. For instance, *moral* norms (m-norms) are those such as,

one shall not steal in normal circumstances,

and *prudential* norms (p-norms) are those such as,

one ought to maximize one's expected utility.

Tuomela argues that norm-obeying means acting for the right normative reason. That is, r-norms are obeyed either because they represent a law, because they represent an agreement, or due to the presence of sanctions (*r-sanctions*). S-norms are grounded in an agent's beliefs, and are fulfilled because such behaviour is expected by others, and because social sanctions (*s-sanctions*) may be applied if they are not. M-norms are obeyed because conscience demands it, and p-norms because it is rational to do so. Now, in the case of conflicts among norms, priorities among them are considered. In general, r-norms override s-norms, but can be overridden by either m-norms or p-norms.

Tuomela also observes that when a group lacks a specific kind of norm, other kinds of norms arise. Thus, a lack of s-norms in force is compensated for by the creation of r-norms (to the enjoyment of lawyers). For example, if the norm

do not commit fraud

is not grounded as a belief in the group of agents, (i.e. it is not a s-norm) it must be issued as a law (or r-norm) whose compliance is monitored and penalised by legally recognised authorities.

2.5.2 Law and Norms

Ross [146] distinguishes norms from directive utterances because whereas directives are just linguistic phenomena, norms are related to social facts. He also distinguishes norms from conformity or patterns of behaviour because when a norm is violated, a social reaction follows. This reaction comes either from individuals acting spontaneously, or from institutionalised organs of the society created for this purpose, such as police, courts, and executive authorities. A fundamental condition for the existence of norms is that, in the majority of the cases, they are fulfilled by the members of a society.

In addition, Ross argues that a norm is *binding* when it arouses feelings of obligations, or when agents feel in a position of coercion such that the norm must be complied with. In fact, compliance with norms is generally enforced by the threat of punishment, which means that there must be a reaction when a norm is violated. Consequently, norms that specify which punishments must be applied against whoever violates a norm must also exist.

Under these considerations, Ross considers norms as directives (or commands) related to certain social facts. He also argues that a norm describes the patterns of behaviour that must be followed by the members of a society. Members, in turn, must feel bound to the norm, and its violation must be penalised. Ross states that a norm includes the following elements: the subjects of a directive, the situations in which the norm must be followed, and the theme of the norm that specifies how subjects must act under the specified conditions. In this way, norms represent obligations for agents.

Ross also defines commands and prohibitions by using obligations as follows. A *command* is a norm that creates an obligation to behave according to its theme. A *prohibition* is an obligation not to behave in accordance with the theme of the norm. Ross defines *norms of conduct* for humans in terms of obligations and prohibitions as follows. If a person A is under an obligation, to another person B, to behave in accordance with the theme of a norm, then B is entitled to *claim* that A must behave in such a way. In other words, a person is entitled to require the other to comply with the norm. In similar terms, the *permission* of person A for person B not to do something means that person A cannot claim that person B must do it.

Norms of competence are also identified by Ross. They explain how new valid norms may be created through the performance of legal acts. *Competence* is a relation among two people, stating that one person is under the obligation to obey the norms created by the other in a correct manner. In other words, one person is endowed with competence to issue new norms in a specific field, and the other, that is subject to this power, has the obligation to obey the former. Consequently, the *subjection* of a person towards another means that the first has *competence* over the second. *Immunity* relations can also be defined through norms so that a person can ignore every other person whose powers cannot be exerted over him, and *disability* occurs when powers of competence cannot be exerted.

2.5.3 Discussion

Norms have long been used as mechanisms to limit human autonomy in such a way that coexistence between self-interested people has been made possible. They are indispensable to overcome problems of coordination of large, complex and heterogeneous systems where total and direct social control cannot be exerted. For this reason, their role has been studied from different perspectives. Philosophy, sociology, psychology and, in particular, legal sciences, have progressed far in this respect, so that they have much to contribute. However, a direct translation of theories in these fields to the field of agents and multi-agent systems is not possible because, in many cases, they are described using natural language which introduces vagueness and ambiguities. These undesirable characteristics are, in general, avoided in computer science by introducing formal methods to specify and verify computational components and, consequently, to produce applications less prone to errors.

We believe that Tuomela's and Ross's research may provide the basis from which a framework to represent norms and normative systems can be created. Their work also sets up the basis to recognise the validity of norms, and explains some of the reasons for agents to fulfill norms. To take advantage of these and other studies of norms, we need to find a means to integrate some of these concepts in models of agents, enabling them to reason about norms. However, one must be aware that there are issues concerning norms which, although interesting, cannot be represented currently. For example, the sense of guilt when a norm is not followed as in the case of moral norms, or the emotions that some humans have in punishing offenders even if this might be costly (and therefore irrational) for them [73, 156], cannot easily be incorporated in existing models.

2.6 Norms as Patterns of Behaviour

An interesting perspective on norms takes them to be desired patterns of behaviour in a group of agents [5, 11, 87, 167, 171]. This view has its origins in game theory, which can be viewed as an extension of decision theory. That is, decision theory is concerned with an isolated agent that must take decisions under conditions of risk and uncertainty. By contrast, game theory deals with decisions in situations of social interaction. These decisions require a strategy of interaction in which the best choice of each participant depends on the actions of others. Thus, each participant knows that the other's actions depend on its own decisions. Both selected strategies and agent interactions converge to patterns of behaviour for a large number of agents over a period of time. These complex patterns of behaviour are known as norms.

In this view, norms are taken to be solutions to problems posed by certain types of social interactions. Ullmann-Margalit [167] identifies three types of situational problems.

- In *prisoners' dilemma* type situations [79, 145], a state of the system is desired by all the participants, but there is also a strong temptation for each to deviate from that state, and the system state that results when all participants deviate is bad for all. The problem here is to devise a method that protects the desired state, and inhibits the temptation to deviate [85].

- In *coordination* type situations [149, 150, 171], there are several mutually beneficial states, none of which is strictly preferred. There is perfect (or almost perfect) coincidence of agent interests. However, there is no possibility of the participants coming to an explicit agreement. The problem is then to find a mechanism that enables them to coordinate their choices of action in order to achieve the desired state.

- Finally, in *inequality* situations, the state of inequality is not completely stable because it is in constant threat. The problem here is for the participants favoured by this inequality to determine how to fortify the state against upsetting the other less favoured participants. In other words, the problem is how to maintain their favoured or powerful position.

2.6.1 Discussion

Much of the research in multi-agent systems has focused on the solutions to problems of coordination, or what has been called *the emergence of norms* that can be beneficial for the system as a whole. This can be seen as a problem of finding the correct strategies that enable agents to converge to situations that are beneficial for all. Having found such a strategy, it becomes a norm for all the members in it, and since this norm is agreed by all agents, it is always complied with by all agents. Although interesting, this approach is not useful for the aims of our work, because rather than being concerned with the process of how norms are created by agents, our research focuses on the *role* of norms and how norms affect the behaviour of agents. As a result, we will not consider this approach further.

2.7 Norms as Constraints for Actions

2.7.1 Social Laws

According to Moses, Shoham and Tennenholtz [127, 153], *social laws* are constraints on the behaviour of agents, and they specify which of the actions that are in general available to agents are allowed in a given state. Shoham and Tennenholtz define *constraints* as pairs composed of an action and a logical proposition that can be true or false in different states. Thus, when an agent is in a particular state and the proposition is satisfied, the action cannot be applied. They define a *social agent* as a tuple comprising a set of actions, a first-order logical language to describe sentences, a set of possible states of the agent, a set of social laws or constraints, and a transition function which, given a state, an action and a set of social laws, provides a set of possible next states for the agent. Such transition functions are used to create plans that satisfy the restrictions imposed by the social laws. In other words, agents are endowed with a set of norms that state what actions must be avoided in predetermined situations. Here, agents are always normative in the sense that they always follow all the restrictions that are imposed on them.

Briggs and Cook extend this model of norms by proposing what they call *flexible social laws* [17]. In their model, agents prefer to obey laws but are able to relax them. Briggs and Cook assume the existence of different sets of laws, ranging from the most strict to the most lenient. In this way, a hierarchy of sets of laws is defined. Then, in trying to achieve goals, agents make plans that fulfill the most strict set of laws. If no plan can be made by following that set of laws, agents use the next set of laws in the hierarchy. Agents continue changing sets of laws, until they find a set that allows them to create a plan to achieve their goals.

2.7.2 Permitted and Forbidden Actions

In contrast to constraints, some people have considered an agent's *rights*. For instance, Norman et al. [133] define a right as an action that can be executed by an agent without being at risk of being penalised by other members of the society. Thus, a right is an action that an agent can legally perform because it is either an inherent property of the agent in the system, or because another agent has permitted it to do so. Agents that cannot achieve their goals due to some restrictions over their actions, must require *permissions* to perform them. Norman et al. define *agreements* between agents as combinations of actions to be performed and the corresponding rights to perform them. There is also a relation that *binds* one or more agents with an agreement, which expresses that agents must defend that agreement. *Commitments* are agreements among two agents, and all agents bound to a commitment, are responsible for defending it. A *moral agent* is also defined as an agent that will not perform an action if it does not have the right to do it.

Alonso [3] defines a right as a permission to perform a set of actions under certain constraints. He argues that, in a group of agents, no other agent is allowed to execute any action that inhibits the rights of an agent, and also that the group is obliged to prevent any inhibitory action. That is, agents *have the right to be protected* from the actions of other agents and, consequently, the notion of group is a guarantee that agent rights and obligations are observed through sanctions and rewards. Unlike Norman et al. Alonso argues that rights can only be exerted until certain conditions become satisfied. He also defines *prohibitions* as those actions that inhibit the rights of other agents, and *obligations* as actions either to prevent or to penalise the violation of a right.

2.7.3 E-institutions

Electronic Institutions are multi-agent systems in which the interactions that take place between agents are regulated by norms [144] and achieved through message interchange. Each message, except the initial message of a conversation, is issued as an answer to a previously issued message. In these systems, norms are used to constrain the kinds of messages an agent can issue in a determined state of a conversation.

Esteva et al. [68, 69, 70], for example, identify four basic elements to define an electronic institution: a dialogic framework, scenes, the performative structure and norms as follows .

- The *dialogic framework* defines the valid illocutions (types of messages) that agents can exchange, the roles of the participants and their relationships.

- *Scenes* are patterns of the conversation between agents in a particular context, and model the dialogues that can take place in a particular activity.

- Dialogues of complex activities are specified by establishing relationships among scenes called *performative structures*, which indicate the role that an agent must play to be able to enter a scene. In this way, the performative structure defines the movement of agents between one activity and another, and scenes define the dialogues that can take place in each activity.

- Finally, *norms* define the obligations of participating agents. Obligations are illocutions that an agent must utter in a specific scene, and *norms* are all the obligations that must hold when a set of illocutions has been uttered, a set of constraints has been satisfied and a second set of illocutions has not been uttered. That is, norms are activated when certain messages have been issued and certain conditions in the environment hold.

Esteva et al.'s work has been used in the implementation of a framework, called *ISLANDER*, which can be used to specify and verify electronic institutions so that designers can check, for example, if all dialogues (scenes) have an initial and end state, if all the defined norms can be activated in one of the defined scenes, and so on.

2.7.4 Discussion

The idea behind social laws is to reduce the options that agents have in a specific state. By using social laws, agents are internally *forced* to find a solution in the way designers want. Social laws are built-in norms, and agents always comply with them. Therefore, neither the concept of authority, nor the idea of being enforced by others are considered. Although the use of social laws allows designers to avoid conflicts among agents, it does not allow unexpected situations in which agents must react in different ways and, therefore, possibly violate a norm; nor does it allow situations in which new norms must be issued and existing norms must be abolished. The same considerations apply for flexible social laws, because although agents can dismiss norms, no consideration of the consequences of doing so are made by any agents. In fact, the social character of norms is not considered in these models because there are no mechanisms to enforce compliance with norms. Moreover, the models do not allow the representation of norms whose compliance might benefit other agents in the society.

By interpreting rights as permitted actions that cannot be inhibited by other agents without the risk of being penalised, both Norman et al. and Alonso recognise the social character of norms. However, although rights seem to work in small groups of agents, their applicability to complex organisations and societies is not clear. In addition, these models do not consider other kinds of norms, such as obligations to do something, or norms that are followed because of social pressure. In these models, it is taken for granted that the group (as a whole) applies punishments, and remains vigilant with respect to all agents complying with norms. However, we argue that, as a matter of practicality, someone must be responsible both for monitoring compliance with norms, and for the application of punishments when compliance does not occur. We also argue that in the same way that agents must recognise whether an action is legal or not, they must also be able to recognise who has authority in the group.

E-institutions are a clear example of the utility of norms and the need for agents that can reason about norms. Agents join these kinds of societies as a way to satisfy their goals, but they must also respect the norms of the society in order to do so. However, norms are different in each institution and, therefore, agents must be able to adopt new norms. In addition, since it is possible that more than one institution can satisfy an agent's goals, agents must be able to decide which institution is better in that respect.

2.8 Social Commitments

Social commitments are considered to be agreements between agents to do something in the future. They provide a certain degree of predictability of agent behaviour because commitments specify not only what must be done, but also under which circumstances and by whom. Thus, commitments are an essential aspect of achieving coordination among a group of agents. Jennings argues that all coordination mechanisms can be reduced to (joint) commitments and their associated (social) conventions [94], and that two kinds of commitments can be created, namely *individual* commitments (or commitments to oneself) and *joint* commitments (which involve more than one agent). All agents involved in a joint commitment must be aware of it and, for this reason, a joint commitment is considered as a shared mental state. In addition, Jennings argues that commitments must be monitored, through their associated conventions, in order to decide whether they are still valid in changing circumstances. So, conventions describe circumstances under which an agent should reconsider its commitments, and indicate the appropriate course of action to either retain, rectify or abandon the commitment.

Castelfranchi [22] also provides a view of commitments which, he says, are closely related to norms and obligations among agents. He identifies three types of commitments: *internal*, *social* and *collective*. Individual commitments correspond to Cohen and Levesque's notion [34] referring to a relationship between an agent and the actions that are performed when an agent decides to do something. Social commitments are created when an agent decides to perform an action for another agent. (Here there is always a third agent that plays the role of a witness). Finally, a collective commitment is the internal commitment of a group of agents. According to Castelfranchi, a social commitment always includes normative elements because an agent agrees to perform an action for another that acquires the right to control and monitor what the first has promised. It also has the right to complain and protest if the first does not perform the action. In addition, collective commitments are created to achieve a common goal, and can be expressed as a set of social commitments where an agent has a commitment with a group, which acquires the rights to monitor the fulfillment of such commitments.

2.8.1 Discussion

Social commitments are a very important concept for any model in which agreements among agents must be reached. This is crucial for systems of autonomous agents in which neither cooperation, nor compliance with previous agreements among agents is guaranteed. In particular, we agree with Jennings and Castelfranchi that social commitments represent a confirmation that what has been promised will be fulfilled. Social commitments imply responsibilities for agents, and social pressure is exerted to make agents fulfill them, which suggests that social commitments can be considered as particular types of norms and, therefore, they are important for this thesis. However, unlike other kinds of norms which persist longer and are recurrently considered, such as obligations in a society, the persistence of social commitments is limited to their fulfilment, i.e. social commitments disappear as soon as agents comply with their promises. Given their importance, we must find the means to incorporate social commitments to a general model of norms.

2.9 Norms as Mental Attitudes

2.9.1 Normative Agent Behaviour

Conte and Castelfranchi [38, 39] state that a norm is a mental notion that establishes actions that ought to be performed by a set of agents. They argue that a norm has two sides: the internal or mental side that corresponds to the agent, and the external side that corresponds to the society. The external side of a norm concerns the process of spreading norms in the social system, or the route a norm follows from legislators to addressees. By contrast, the internal side of a norm is related to its internal representation, and to all the processes that occur inside the agent in order to adopt or comply with a norm. Conte and Castelfranchi state that norms are aimed at controlling the behaviour of agents subject to them, and that this control is possible because when agents receive a norm, they create a *normative belief*, which represents a belief about an obligatory social requirement. From these beliefs, new goals are generated in the mind of addressee agents. These kinds of goals are called *normative goals*.

In addition to such mental concepts, Conte et al. [43] discuss two decision-making processes concerning norms: the *acceptance of a norm* and the *decision to conform to it*. To accept a norm, agents must be able to evaluate candidate norms against several criteria. For instance, a norm must be rejected if it is an instantiation, application or interpretation of another norm, if the agent that issues the norm is a non-recognised authority, if the norm is not directed to the agent itself, or if addressee agents are not within the scope of an authority. Furthermore, an agent will only accept a norm if by doing so some of its goals are satisfied in the future.

Conte et al. state that once a norm is accepted and the corresponding normative goal formed, the decision to comply with it is made based on several factors. For instance, a normative goal is dropped if there is a conflict with goals that are more urgent than it. In this case, agents must reason about the expected value of the violation of a norm, which depends on several factors such as the probability and weight of punishments, the importance of the goal, the value of respecting the norms and being a good citizen, the importance of possible feelings related to norm violation (guilt, indignity, etc.), and the importance of foreseen negative consequences of the violation for the global interest. A norm can also be violated when there is a conflict with other norms already adopted, when the agent believes that the norm is not its concern, or when the norm prescribes an action that cannot be executed.

2.9.2 Normative Agent Models

Dignum et al. [54] present a modified BDI-interpreter to deal with norms and obligations. They state that norms are different to obligations, because whereas the objective of *norms* is to make the behaviour of agents standard in order to facilitate the cooperation and interaction of agents within a society, *obligations* are associated with specific enforcement strategies that involve punishment for their violators. According to Dignum et al., norms are beneficial for the group, there are neither punishments nor rewards for complying with them, and they are followed as an end. By contrast, obligations are fulfilled whenever there is a probability of being caught, and the cost of punishment is higher than the cost of adhering to such an obligation. Norms and obligations are beliefs and, in that sense, an agent may have an incomplete or incorrect understanding of them. In Dignum et al.'s model, agents order their norms based on the preferences of the social benefits of a particular situation. Conversely, the obligation preference order is based on the cost of punishment when an obligation is not fulfilled.

To include reasoning about norms and obligations, Dignum et al. modify a BDI architecture to identify *deontic events*. These events determine which norms and obligations must be applied. That is, they represent invocation conditions for a set of plans that must be considered in order to fulfill the corresponding obligation or norm. These active plans are fed into a deliberation process which determines which plan must be executed based on the preferences for norms and obligations mentioned above.

Boella and Lesmo [12] present another proposal for agents able to reason about norms. They consider a norm as an obligation that involves at least two individuals (modelled as intelligent deliberative agents): the *bearer* of the obligation that must respect the norm, and the *normative* agent (or authority) that wants the norm to be fulfilled. This authority also has the right of imposing punishments to offenders of a norm. In their work, *obligations* are considered as 4-tuples which include: the content of the obligation, its bearer, a normative agent, and an action (which they call sanction) that the normative agent will bring about in the case of detecting the violation of an obligation.

Boella and Lesmo ground their agent architecture on *situated* BDI agents that choose one of a set of potential plans to perform. A utility function is also included to evaluate the outcomes of actions, and to help agents to select a plan that maximises their expected utilities. To introduce reasoning about obligations, Boella and Lesmo modify the architecture of these agents as follows. First, the planning phase considers the agent's obligations in order to avoid forbidden actions. Then, the plan selection phase, besides considering the utility function mentioned above, includes a process in which agents simulate the reaction of the normative agent (or authority). In fact, agents analyse the possibility that the normative agent selects, as its next goal, the application of a punishment if the norm is violated. This is possible because agents know that normative agents are also self interested and, therefore, the only way in which a normative agent selects the application of a sanction as its next goal is when the plan associated with such a goal offers greater utility than other available options. Consequently, the decision of fulfilling an obligation is a trade-off between the cost (in terms of time or resources consumed) of doing something for achieving the obligation and the effects of the reaction of the normative agent.

Castelfranchi et al. [29] propose an agent architecture which is not directly based on BDI agents, though the generic architecture that is used as its basis is. In this architecture, a special *maintenance-of-society-information* module is included, which is responsible for both accepting and storing the norms that are directly extracted from the communicated information. Another related component is the *norm manager*, which determines which norms the agent adopts or rejects and, on the basis of these norms, creates some meta-goals. These meta-goals are passed to the *strategy-management* component to determine the strategies used in the creation and selection of goals and plans.

2.9.3 Discussion

Conte and Castelfranchi's work on norms contributes towards explaining the role of norms and identifying some of the processes of deliberation regarding norms that agents undertake. However, their work is more intuitive than formal, and gaps and ambiguities can be found in it. For example, neither the way in which normative goals are generated from norms, nor the way in which other processes of decision are affected is mentioned. Moreover, although they mention some criteria used by agents to decide whether to accept and comply with norms, the way to do this is not described.

Regarding models of normative agents, the most important contribution of the work described above lies in the acknowledgment that agents must be provided with the means to deliberate about when and why they must fulfill a norm. However, these models address the problem only partially. Both Dignum et al. and Boella and Lesmo describe specific strategies for decision-making, the first based on the cost of complying with norms, and the second based on the intentions of agents responsible for applying punishments. In both models there is no indication of how other current goals and intentions may be affected by any decisions that agents take regarding norms. Their model is restrictive and agents that follow other strategies to comply with norms cannot be represented. Although the architecture provided by Castelfranchi et al. is more general, and they mention that norms must affect the processes of selecting goals and plans, they do not consider the problem further. Consequently, we consider that in order to accommodate the richness of norms into agents and multi-agent systems, a more general model for normative agents must be provided.

2.10 Modelling of Normative Concepts

2.10.1 Categories of Norms

There are neither common agreements about the structure of norms nor about the different kinds of norms that can be used in multi-agent systems. However, some work has already been done towards the unification of normative concepts. Dignum [52, 53], for example, divides norms into three levels: the *private*, the *convention*, and the *contract* levels.

At the private level, norms are expressed as *preferences* that allow agents to make private judgements between different obligations or goals, in order to determine which actions agents will take. In other words, when there is an obligation or goal that must be satisfied, an agent might prefer certain situations to be true. For instance, if an agent has to travel, it would prefer to travel free.

According to Dignum, the convention level of norms provides a kind of *moral background* for agents to interact. Conventions are generally fixed when the system is initiated. There are two kinds of convention: *interpretation rules* and *prima facie* norms. Interpretation rules are used to indicate how terms must be interpreted by the agent. For example, they can be used to explain what "*reasonable*" or "*cheaper*" mean. They can also indicate the implicit effects that the execution of one action may have. For instance, a rule can state that when a good is bought, it must be paid for. Dignum says that prima facie norms are general social norms and values that can be given as prohibitions or permissions, and that *prohibitions* are limitations on the behaviour of agents, and *permissions* are used to indicate exceptions to a general rule in cases of uncertainty.

Contracts are defined by Dignum as sets of *obligations* and *authorisations* between agents, and a *directed obligation* means that an agent is forced either to perform an action or to maintain a situation for other agents. All these concepts belong to the level of contracts. Dignum also states that an *authorisation* describes the obligation from the point of view of the other agent. That is, the other agent has authorisation to demand the fulfillment of an obligation, as well as authorisation to claim compensation in case the obligation is not complied with. Consequently, contracts describe the types of relation that hold between agents and their mutual expectations of behaviour. They have a specific objective, and hold for a limited period of time (until the objective is satisfied). Dignum states that by using norms in this way, legal contracts, cooperation, and informal agreements between agents can be easily described.

Singh [157] presents a framework called *spheres of commitments* where agents can be recursively composed of heterogeneous individuals or groups of agents. A sphere of commitments is a group of agents, together with its roles and its concomitant commitments. His framework defines commitments and some operations over them as follows. A commitment represents an agent compromising itself to bring about a situation for another agent that belongs to the same group of agents. A commitment can be created, discharged, cancelled, released, delegated, or assigned. Operations for groups are considered. For instance, a group can be created, an agent may adopt a role, an agent may re-assign itself to another role, or an agent may exit a group.

Singh distinguishes two kinds of commitment: *explicit* and *implicit*. Explicit commitments are created after direct interaction between two or more agents, while implicit ones simply represent common knowledge in the system. In addition, Singh defines *social policies* as restrictions over the kind of operations that can be performed on a set of commitments.

According to Singh, by defining relationships between commitments and operations, different normative concepts can be defined as follows. *Pledges* are explicit commitments arising from commissive performatives, in which all commitment operations are permitted. *Ought* commitments are those among the members of a group in which operations to cancel, delegate or assign are not permitted. *Taboos* are implicit commitments that can neither be cancelled nor overridden by other commitments. *Customs* or *conventions* are implicit commitments that that can neither be cancelled nor overridden by other commitments but can lead to other commitments. *Collective* commitments are the conjunctions of the commitments of the individuals to the group. In Singh's view, *obligations* can be either pledges or *ought* commitments.

By using his definition of commitments, Singh also provides definitions for traditional concepts as follows. *Claims* are what agents can demand from others and, therefore, they are defined as commitments. *Privileges* represent the freedom agents have from the claims of others, and *power* refers to the ability of an agent to force the alteration of a legal relation. Finally, *immunity* means freedom from the power of another agent.

2.10.2 Normative Reasoning

Deontic logic refers to the logic of invitations, requests, commands, rules, law, moral principles, and judgments. For this reason, it has been used for a long time in the representation of reasoning about legal matters, i.e. to represent the way that human beings ought to behave according to the normative principles that drive them. Currently, its use in agents and multi-agent systems seems to be justified with the introduction of norms [173] because, by using deontic inferences, the ways in which agents must behave can be represented [107, 168, 169].

In contrast to propositions, norms do not have a truth-value and, consequently, propositional calculus cannot be used to make inferences about them. *Deontic logic* was created with this objective in mind. Instead of assigning truth-values to norms, deontic logic uses the concept of *validity* of norms. Then, by defining operators on norms (similar to those used in propositional logic such as *and*, *or*, and *not*), inferences on deontic events are made. Until now, different kinds of deontic logics have been proposed to overcome different problems. For example, some but not all deontic logics allow the representation of the so called *contrary-to-duty* norms, which specify obligations that are in force only in sub-ideal situations.

One of the main contributions in this field is the work of Jones and Sergot [101, 102, 151], which is directed towards the application of deontic logic both for the construction of legal expert systems able to analyse legal text, and also for the formal specification of institutions whose members are controlled by norms. They state that, at the appropriate level of abstraction, law, computer systems, and many other kinds of organisational structures may be viewed as instances of normative systems. Barbuceanu et al. [8, 9] also use a variant of deontic logic to enable agents to reason about forbidden and obliged goals based on the cost of complying with their obligations.

2.10.3 Discussion

The classifications of both Dignum and Singh help us to understand the different kinds of norms that agents must deal with, and although we do not completely agree with some of their definitions, there are many interesting points that deserve our attention. For example, by defining contracts as pairs composed of obligations and authorisations, Dignum highlighted the importance of those norms that specify what must be done when an obligation is not fulfilled. Singh's perspective is also very interesting because he shows how, by defining a single normative concept, the most common normative terms can also be defined. In addition, he notes the importance of contextualising norms and agents into a specific group.

One of the major problems that we observe in Dignum's classification is that each category seems to be very different from others and, consequently, agents would have to apply a different process of reasoning for each one of them. This may complicate any model of normative agents. In addition, neither Dignum nor Singh provide a model for norms that allows agents to reason about *why* a norm should first be adopted and then complied with.

Now, although we recognise the importance of deontic logic to represent knowledge and reasoning about the normative behaviour of agents into systems regulated by norms, its use does not address some important issues for this thesis. In particular, autonomous decisions of agents are difficult to model by using deontic logic because deontic logics deal with things that are obligatory for agents but not with things that are desired by agents, which is the source of many conflicts of interest. We argue that, sometimes, autonomous agents must decide what is more important, their social responsibilities or their own goals, and since this problem is not easy to represent in deontic logic, we will not consider its use further, but will examine alternative formalisms.

2.11 Conclusions

To sum up the main points highlighted in this review chapter, we start by saying that motivations are the key to understanding the decisions of autonomous agents. In particular, motivations enable agents to take decisions when conflicting situations are found and, since the coexistence of self-interested agents in general causes conflicts, motivations must be considered to explain the behaviour of self-interested agents that must interact with other agents of similar characteristics.

The BDI model of agents has been applied with great success in different domains. It has been taken as the basis to develop other agent models with additional characteristics such as those that allow agents to interact with other agents in the environment. We consider that the model can also be enhanced to cope with the presence of norms. This can aid the work of agent designers who can reuse previously designed components to model normative agents.

Being autonomous does not mean being asocial. Autonomous agents work to satisfy their own goals but they can still cooperate with other agents or they can even join societies. Thus, although their autonomy sometimes becomes constrained, autonomous agents are able to adopt and fulfill the norms of a society. To provide an effective model of autonomous and normative agents, we have to explain the reason agents have either to create relationships with other agents or to join societies (and to adopt and comply with their norms). Powers and dependence are some of the explanations for these decisions, and although they were initially considered as relationships that emerge due to an agent's capabilities, they can also be explained as a result of the roles agents play in a society.

Advances in the study of norms from the view of many social sciences can be exploited to create models of norms, models of agents able to reason about norms, and models of multi-agent systems that are regulated by norms. However, theories from other disciplines must be coupled with current successful models of agents in order to facilitate the incorporation of these new characteristics into previously developed models of agents and multi-agent systems. In addition, for a concept to be incorporated into the agent field, it must be well defined and formalised in order to facilitate their computational implementation.

In general, the problem of modelling agents able to reason about norms is far from trivial, and although some research has been done, more is needed [44, 45]. Besides finding a model that describes the normative behaviour of autonomous agents, problems regarding other issues must also be faced. In particular, the following problems require an immediate answer.

- There is no canonical representation of norms. Although the majority of views agree that norms prescribe patterns of behaviour for a set of agents, and that there is social pressure to enforce them, there is neither consensus about their meaning, nor about the components that norms must include.

- There is no consistent way to reason about different kinds of norms. Several categories of norm have been already proposed; however, instead of facilitating reasoning about them, this causes confusion. They give a false indication that many reasoning process must be implemented for each kind of norm, which makes agent models much more complex.

- To comply with norms, agents must recognise themselves as part of a system and, consequently, agents must have a model of the system they are in. Given that systems regulated by norms have been designed with the objective of making them efficient, many of the elements needed by autonomous agents to take decisions regarding norms have been ignored. In particular, there are no means to limit the authority of some agents and, therefore, agents are condemned to obey them forever without the possibility of leaving or staying in a society.

- In current research on norms, no distinction is made between norms as abstract specifications and norms as mental attitudes from which goals for agents are derived. Without considering these differences, norms being adopted, fulfilled, or violated by agents cannot be represented in a model for norm alive reasoning.

By taking the perspective of individuals agents, many of the gaps in current models of norms, and systems regulated by norms, can be filled and, more importantly, a general model for agents able to reason about norms and powers can be proposed.

Chapter 3

Grounding the Theory of Normative Agents

3.1 Introduction

Since many concepts regarding agents do not have a common meaning [120], before introducing our theories, it is necessary to provide a vocabulary that includes definitions for agents and the mental attitudes that determine their behaviour. However, describing concepts by only using natural language might introduce ambiguities because natural language is vague and imprecise, and this can lead to severe problems not only in the theories but also in any application of them [57]. Two problems arise here. On the one hand, definitions for agent concepts that act as building blocks to develop consistent theories of norms, agents that reason about norms, and multi-agent systems that are regulated by norms, must be provided. On the other, the means to describe in a precise and unambiguous manner, not only the basic concepts regarding agents but also the new concepts introduced in this thesis, are needed. Both problems are the main concern of this chapter.

In providing a common vocabulary of precise and unambiguous terms, *formal methods* play an important role [170]. Formal methods are mathematical modelling techniques used in the specification and design of computer systems. They allow inconsistencies, ambiguities and incompleteness in a specification to be opportunely detected [33]. Whereas specification is the process of describing a system and its desired properties, formal specifications do so by using languages, called *formalisms*, with a mathematically defined syntax and semantics [175]. So, precision and understanding of agent theories can be increased if a mathematical basis is used in conjunction with natural language.

Introducing formal specifications to describe theories provides many advantages because they can be mechanically checked and, above all, they allow the properties of a described system to be verified. Some agent researchers have used different logics as formalisms to express their theories of agents, for example temporal logic, modal logic, and deontic logic. However, designers have had difficulty in implementing them because the logics are not oriented to creating software systems. Since one of the principles of this thesis is to provide theories yielding software implementations, we need formalisms that facilitate this work. One of these formalisms is the language Z, which is adopted here to describe our theories.

Now, there is much work that formally describes properties and terms related to agents [81, 142, 152], and work that goes beyond this to formally describe agents and multi-agent systems [62]. Thanks to them, it is not necessary to start the labour of defining every basic concept regarding agents and multi-agent systems from scratch. These frameworks provide principles and well-defined terms that can be used as the foundations for more sophisticated theories. In this thesis, we adopt the SMART agent framework [62] as the basis to develop our theory of norms mainly because its concept of *motivations* as the driving force that affects the reasoning of agents in satisfying their goals, is considered here as the underlying argument for autonomous agents to reason about norms. The SMART agent framework is also adopted because it describes how and why important relationships between autonomous agents emerge as a result of agents voluntarily satisfying the goals of other agents. This is important because norms are social concepts that prescribe the satisfaction of some goals from which some agents might benefit and, therefore, norms are also the means to relate agents. However, since SMART is intended to cover a wider range of agents and multi-agent systems, we must refine many of its concepts in order to make them suit our purposes.

To satisfy the objectives of this chapter, four more sections are introduced. First, in Section 3.2, the reasons to adopt the Z language in this thesis are given, and the language itself is described. After that, the concepts of the SMART agent framework that are important for our theories are mentioned in Section 3.3. Agents and the basic mental attitudes that determine their behaviour are defined in Section 3.4, whereas autonomous agents and motivations are defined in Section 3.5 before concluding.

3.2 Notation

3.2.1 Introduction

The formal specification language Z is a mathematical language based upon set theory and first order predicate calculus. Z extends the use of these languages by allowing an additional mathematical type known as a *schema* where objects and their properties are put together. Schemas are a powerful structuring mechanism because new and more complex schemas can be defined by using previously well-defined schemas. Together with natural language, the Z specification language enables the provision of specifications in the strict sense of software engineering [162]. That is, Z specifications are easy to read and understand, and the language allows incremental building of complex specifications, reusability and a smooth transition from specification to implementation through well defined techniques of refinement. Thus, a specification can be refined into another that is closer to executable code [175]. Moreover, since in Z every object has a unique type, type checkers can be used to check specifications and detect inconsistences, ambiguities, and incompleteness. In particular, all the specifications provided in this thesis have been type-checked by using the type-checker *fuzz* for Z.

The Z language can be used to describe the state of a system, and the ways in which that state may change. This property makes the language useful to describe agents since they are situated in an environment and any action they perform might change such an environment. The effectiveness of the Z language to specify agents properties has been demonstrated by Goodwin in [81] and by d'Inverno and Luck in [58, 59, 62, 120] among others. There are, however, some concerns about the effectiveness of Z to model agent interactions because Z is not intended for the description of timed or concurrent events. In these cases, Z can be used in combination with other formal methods that are well suited for those purposes.

In this section, rather than providing a detailed description of the Z language, general descriptions of the elements of Z that are used in this thesis are given. A summary of these elements is shown in Figure 3.1, and more information about the language and its use can be found elsewhere [163, 175, 179].

3.2.2 The Z Specification Language

As mentioned before, the Z language is based upon set theory and a first order predicate calculus, hence, concepts such as set operators (i.e. union, intersection, etc.), cartesian product, power sets, logical operators, and universal quantifiers are used to describe object properties. Now, to introduce a basic type in Z, the notion of a given *set* is

Definitions and declarations

a, b	Identifiers
p, q	Predicates
s, t	Sequences
x, y	Expressions
A, B	Sets
R, S	Relations
$d : T$	Declarations
$a == x$	Abbreviated definition
$[a]$	Basic type definition
$A ::= b\langle\langle B\rangle\rangle$ $\mid c\langle\langle C\rangle\rangle$	Free type definition
$\mu\, d \mid P$	Definite description
let $a == x \bullet y$	Local variable definition

Logic

$\neg\, p$	Logical negation
$p \wedge q$	Logical conjunction
$p \vee q$	Logical disjunction
$p \Rightarrow q$	Logical implication
$p \Leftrightarrow q$	Logical equivalence
$\forall\, d : T \bullet q$	Universal quantifier
$\exists\, d : T \bullet q$	Existential quantifier

Sets

$x \in y$	Membership
$x \notin y$	Non-membership
$\varnothing$	Empty set
$A \subseteq B$	Set inclusion
$\{x, y, \ldots\}$	Set of elements
$(x, y, \ldots)$	Ordered tuple
$A \times B \times \ldots$	Cartesian product
$\mathbb{P}\, A$	Power set
$\mathbb{P}_1\, A$	Non-empty power set
$A \cap B$	Set intersection
$A \cup B$	Set union
$A \setminus B$	Set difference
$\bigcup A$	Generalized union
$\# A$	Size of a finite set
$\{d : T \ldots \mid p \bullet x\}$	Set Comprehension
first p	First of pair
second p	Second of pair
min A	Minimum of a set
max A	Maximum of a set

Relations

$A \leftrightarrow B$	Relation
$\mathrm{dom}\, R$	Relation Domain
$\mathrm{ran}\, R$	Relation Range
R^{-1}	Relational Inverse

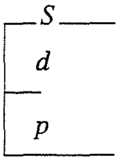
Functions

$A \mathrel{+\!\!\!\!\rightarrow} B$	Partial function
$A \rightarrow B$	Total function

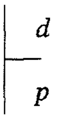
Sequences

$\mathrm{seq}\, A$	Sequence
$\mathrm{seq}_1\, A$	Non-empty
$\langle\rangle$	Empty
$\langle x, y, \ldots\rangle$	Sequence
$s \frown t$	Concatenation
head s	First element
tail s	All but first

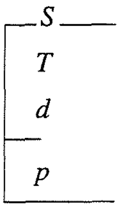
Schema notation



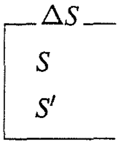
Schema



Axiomatic definition



Inclusion



Operation

$S \widehat{=} [d : T \ldots]$	Schema definition
$z.a$	Schema component
$S \fatsemi T$	Sequential composition
$a?$	Input to an operation

FIGURE 3.1: Summary of Z notation

used. For instance, the sentence [*STUDENT*] may be written to represent the set of all students. If it is desired to state that a variable takes on the set of students, $x : \mathbb{P}\,STUDENT$ must be written, whereas, if the variable is an ordered pair of students, $x : STUDENT \times STUDENT$ is written.

To represent more complex structures, *schemas* are introduced. Z schemas have two parts: the upper declarative part, which declares variables and their types, and the lower predicate part, which relates and constrains those variables. The type of any schema can be considered as the Cartesian product of the type of each of its variables, without any notion of order, but constrained by the schema's predicates. For example, if we want to represent a class which consists of a limited set of students already enrolled, a schema whose declarative part includes a set of students and a variable that represents the maximum allowed number of students, is given as follows.

$$\begin{array}{|l}
\multicolumn{1}{l}{\textit{Class}} \\ \hline
enrolled : \mathbb{P}\,STUDENT \\
maximum : \mathbb{N} \\ \hline
\#enrolled \leq maximum \\ \hline
\end{array}$$

Class is the name of the schema, and its predicate part states that the cardinality of the set of students is always lower than the maximum specified. This constraint must always be fulfilled. Each schema is a type that can be used to define new variables. For instance, the sentences *cs101:Class* and *cs203:Class* may represent two different Computer Science classes. A variable included in a schema can be accessed by writing the name of the schema variable followed by a dot and the name of the required variable. For instance, $cs101.maximum$ is used to refer the maximum number of students allowed in class $cs101$.

Modularity is facilitated in Z by allowing schemas to be included within other schemas. For example, to represent the students in a class who have been evaluated, the *Class* schema can be included in a new schema as follows.

$$\begin{array}{|l}
\multicolumn{1}{l}{\textit{ClassEvaluated}} \\ \hline
Class \\
evaluated : \mathbb{P}\,STUDENT \\ \hline
evaluated \subseteq enrolled \\ \hline
\end{array}$$

Here, all variables and constraints on variables of the first schema are included in the declarative and predicate part respectively in the second schema. Thus, the *ClassEvaluated* schema is equivalent to the following schema.

ClassEvaluatedEq

$enrolled : \mathbb{P}\, STUDENT$

$maximum : \mathbb{N}$

$evaluated : \mathbb{P}\, STUDENT$

$\#enrolled \leq maximum$

$evaluated \subseteq enrolled$

Besides schema *inclusion*, *conjunction* and *disjunction* of two schemas are permitted. These operations result in schemas whose declarative part is taken as the union of the declarative parts, whereas the predicate part represents the conjunction, or disjunction, of the predicate parts of each of the involved schemas. This schema calculus is a method of building new schemas from old ones.

Operations on the variables of a state schema are defined in terms of *changes* on the state of such variables. An operation is denoted by the symbol Δ preceding the state schema on which the operation is performed. Specifically, an operation relates (initial) variables before and (final) variables after the operation. Final variables are denoted by dashed variables. Operations may also have inputs represented by variables with *question marks*, and outputs represented by variables with *exclamation marks*. For instance, to represent the operation to enrol a new student who has not previously been enrolled in a class with a limited number of places, we use the following schema.

Enrolling

$\Delta Class$

$newstudent? : STUDENT$

$newstudent? \notin enrolled$

$\#enrolled \leq maximum - 1$

$enrolled' = enrolled \cup \{newstudent?\}$

In the above schema, the first two predicates are the constraints that must be satisfied before performing the operation; whereas the last predicate represents the results of the operation. When the operation does not change a state schema, it is preceded by the symbol Ξ. *Composition* of operations on schemas is also possible. The composition operator is denoted by the fat semicolon $\fatsemi$, and indicates that the final states of the first operation are taken as the initial states of the second. For example, if an operation which gives as output the number of students already enrolled in a class is required, a schema is written as follows.

$$\begin{array}{|l}
\multicolumn{1}{l}{\mathit{HowMany}} \\ \hline
\Xi \mathit{Class} \\
\mathit{total}! : \mathbb{N} \\ \hline
\mathit{total}! = \#\mathit{enrolled} \\ \hline
\end{array}$$

Now, to represent an operation that enrols a student in a class, and then displays how many students there are, we use the following composition operation.

$$\mathit{EnrollandCount} \mathrel{\widehat=} \mathit{Enrolling} \mathbin{⨾} \mathit{HowMany}$$

A *relation* type expresses some relationship between two existing types, known as the *source* and the *target* types. The type of a relationship with source X and target Y is $\mathbb{P}(X \times Y)$. A relation is therefore a set of ordered pairs. When no element from the source type can be related to two or more elements from the target type, the relation is a *function*. For example, the relation below defines a *function* which relates $n1$ to $n2$, $n2$ to $n3$, and $n3$ to $n2$.

$$\mathit{Rel}1 = \{(n1, n2), (n2, n3), (n3, n2)\}$$

A *total* function ($\rightarrow$) is one for which every element in the source set is related, while a *partial* function ($\rightarrow\!\!\!\!\!+$) is one for which not every element in the source is related. The *domain* (*dom*) of a relation or function comprises those elements in the source set that are related, and the *range* (*ran*) comprises those elements in the target set that are related. In the example above, $\mathrm{dom}\,\mathit{Rel}1 = \{n1, n2, n3\}$and $\mathrm{ran}\,\mathit{Rel}1 = \{n2, n3\}$. The *inverse* of a relation is obtained by reversing each of the ordered pairs so that the domain becomes the range, and the range becomes the domain. A *sequence* (*seq*) is a special type of function where the domain is the contiguous set of numbers from 1 up to the number of elements in the sequence. For example, the relation below defines a sequence.

$$\mathit{Rel}2 = \{(1, n4), (2, n3), (3, n2)\}$$

Sets of elements can be defined in Z by using set comprehension. For example, the following expression denotes the set of squares of natural numbers greater than 10:

$\{x : \mathbb{N} \mid x > 10 \bullet x * x\}$. Functions, relations, and variables can be also defined outside an schema by using *axiomatic definitions*. Axiomatic definitions, like schemas, contain declaration and predicate parts. All elements defined through an axiomatic definition are considered as global elements that can be used in any subsequent schema. The following example defines a function that gives the square of a natural number.

$$
\begin{array}{|l}
square : \mathbb{N} \rightarrow \mathbb{N} \\
\hline
\forall n : \mathbb{N} \bullet square(n) = n * n
\end{array}
$$

3.3 The SMART Agent Framework

In providing a common vocabulary of terms regarding agents and their behaviour, the use of very well founded frameworks for agents and multi-agent systems avoid the work of starting from scratch and reduce the risk of having inconsistences at low levels. In this thesis, the SMART (Structured and Modular Agents and Relationship Types) agent framework developed by Luck and d'Inverno [62] has been adopted. It contains very well defined concepts regarding agents and their interactions in multi-agent systems that are considered here as the basis for our work on norms.

SMART describes the environment as a collection of entities of four different types which are organised in a hierarchy. *Entities* are things in the environment that can be described by using a set of attributes. *Objects* are entities capable of doing some actions. *Agents* are objects capable of achieving goals, and *autonomous agents* are agents with motivations. Then, motivations are defined as desires or preferences that affect the outcome of any decision-making process. Goals and motivations are crucial to define agency and autonomy because agents can satisfy goals, but autonomous agents also have reasons to satisfy them. Thus, the framework clearly distinguishes between what must be done (goals) from the reasons for which it must be done (motivations). This is very important in our work about norms since we consider that autonomous agents, rather than adopting and complying with norms as an end, have reasons to do so, and that these reasons can be explained in terms of their goals and motivations.

In SMART the concept of agent *interaction* is key to defining multi-agent systems. It explains interaction as a result of an agent satisfying the goal of another. Then, *multi-agent systems* are defined as systems that contain two or more agents where at least one of them is autonomous. In these systems, there is at least one relationship between two agents one of which is satisfying the goal of the other, and from which interaction between agents emerges. SMART describes how, through adoption of goals, different relationships between agents arise in a multi-agent system. Particularly important for this thesis are the relationships of *cooperation* where autonomy plays an important role in distinguishing between doing something as an end or as a voluntary decision. Thus, for agents to adopt a goal, they must be convinced rather than being imposed. So, autonomous agents cooperate when they have voluntarily agreed to adopt the goal of other agent.

Being able to identify the relationships in which they are involved, allows agents to exploit the capabilities of others to achieve their own goals. This capability is important when agents have to evaluate alternative plans in which other agents might be affected. Voluntary relationships are key for modelling agents able to decide on their own whether to enter and remain in a society (and, therefore, to comply with its norms) is important for their own goals.

Since the framework is intended to cover a wide range of agents, an internal agent architecture is not prescribed. Nevertheless, Luck and d'Inverno provide an example of how a specific architecture can be incorporated into SMART [59]. In our case, a BDI-like agent model is also used to describe the normative behaviour of agents, and some of the concepts defined in this chapter have already been defined in SMART. However, new concepts are also introduced. In particular, those concepts that correspond to the relations between goals (Subsection 3.4.4) and the importance of goals (Subsection 3.5.2) are new aspects of our model.

3.4 Agents

BDI is one of the most successful model of agents, and it has been chosen as the starting point towards a model of normative agents. In a BDI model, agents are endowed with different mental attitudes (namely *beliefs*, *desires* and *intentions*) which together with processes to decide what to do (*deliberation* processes) and processes to decide how to do it (*means-ends reasoning*) determine their behaviour [177]. Besides mental attitudes, a definition for agents and some goal relationships are provided in this section. Descriptions of the processes of deliberation and means-ends reasoning can be found elsewhere [15, 141, 176], and the cases in which these processes are affected by norms will be described in detail in subsequent chapters.

3.4.1 Primitives

Agents are situated in an environment that can be described as a set of attributes [117]. An *attribute* is defined as a perceivable feature of the world and can, therefore, be represented as a predicate or its negation. Details of predicate representations are not relevant for this work but can be found elsewhere [58, 59]. Here, predicates are formally defined as given sets, which means that we need say nothing more about them.

$[\mathit{Predicate}]$

Now, the formal representation of attributes is given through a *free type definition* in Z language. It states that attributes are either positive predicates (those preceded by *pos*) or negative predicates (those preceded by *neg*).

$\mathit{Attribute} ::= \mathit{pos}\langle\langle \mathit{Predicate}\rangle\rangle \mid \mathit{neg}\langle\langle \mathit{Predicate}\rangle\rangle$

The *state* of the environment is defined as a set of attributes that describes all the features of the world that hold at a particular time.

$\mathit{EnvState} == \mathbb{P}_1\, \mathit{Attribute}$

Actions are discrete events that change the state of the environment when performed. Then, the set of all possible actions that can be performed in an environment is formally defined as follows.

$\mathit{Action} : \mathbb{P}(\mathit{EnvState} \rightarrow \mathit{EnvState})$

3.4.2 Beliefs

Beliefs are internal representations of the information that one agent has about both itself and its environment. Due to the limited perception of agents, beliefs are not always true; however, they persist until an agent obtains new information that contradicts them. *Beliefs* are formally defined as attributes.

$\mathit{Belief} == \mathit{Attribute}$

3.4.3 Goals

Goals are defined as states of the world that an agent wants to bring about and, although an agent may have several goals, just one will be carried out at one time. Since states of the world can be represented as predicates or their negation, we also use a non-empty set of attributes to formally define goals as follows.

$\mathit{Goal} == \mathbb{P}_1\, \mathit{Attribute}$

A goal is considered as *satisfied* if the state that represents it is a logical consequence of the current state of the environment. Defining if a state is a logical consequence of another is computationally intractable and, although this can be easily done by humans, it can also require a huge quantity of computational work [143]. Dealing with this problem is beyond the scope of this thesis; instead, we abstract it and introduce a new predicate (*logicalconsequence*) which is true when the second argument is a *logical consequence* of the first.

$$logicalconsequence\_ : \mathbb{P}(\mathbb{P}\,Attribute \times \mathbb{P}\,Attribute)$$

The formal representation of a satisfied goal can then be given as follows.

$$\begin{array}{l}
satisfied\_ : \mathbb{P}(Goal \times EnvState) \\
\hline
\forall\, g : Goal;\ st : EnvState \bullet \\
\quad satisfied\ (g, st) \Leftrightarrow logicalconsequence\ (st, g)
\end{array}$$

3.4.4 Goal Relationships

To take effective decisions, agents assess how, by satisfying some goals, other goals might be affected. We start with goals that negatively affect others by defining goals in *conflict*. Sometimes a conflict is easy to observe because the state of one goal is simply the negation of the other, such as being outside a room and being inside it at the same time. In general, however, conflicting situations are more difficult to identify. For example, cleaning a room and watching a favourite TV programme can be in conflict if both activities take place at the same time and in different locations. Formally, we say that a goal *conflicts* with another when the second is not a logical consequence of the first. This relationship is formally represented as follows.

$$\begin{array}{l}
conflicts\_ : \mathbb{P}(Goal \times Goal) \\
\hline
\forall\, g_1, g_2 : Goal \bullet \\
\quad conflicts\ (g_1, g_2) \Leftrightarrow (\neg\ logicalconsequence\ (g_2, g_1))
\end{array}$$

Knowing when a goal is a *subgoal* of another is also important for our aims. Subgoals are, in an intuitive sense, those goals that contribute towards the satisfaction of a goal. That is, a goal contributes to another when the first represents a step towards the satisfaction of the second as, for example, when a tourist buys a flight ticket as a first step towards going her holiday. Following this intuitive meaning, some properties for subgoal relationship can be given as follows [63].

Subgoals are *consistent*; that is, a subgoal cannot prevent its super goal. A goal is a subgoal of itself (the relation is *reflexive*). The subgoal relationship is also *transitive*, so that subgoals of a subgoal, are also subgoals of the super goal. Finally, no goal has an infinite chain of subgoals (the relation is *well-founded*). Formally, the *subgoal* relationship is a consistent relation whose domain and range are defined in the set of all the goals. It is reflexive, transitive, and well-founded, and it is represented as follows.

$$
\begin{array}{|l}
\mathit{subgoal}\_ : \mathbb{P}(\mathit{Goal} \times \mathit{Goal}) \\
\hline
\forall g_1, g_2 : \mathit{Goal} \bullet \mathit{subgoal}\,(g_1, g_2) \Rightarrow (\neg\, \mathit{conflicts}\,(g_1, g_2)) \\
\forall g_1 : \mathit{Goal} \bullet \mathit{subgoal}\,(g_1, g_1) \\
\forall g_1, g_2, g_3 : \mathit{Goal} \bullet \\
\quad ((\mathit{subgoal}\,(g_1, g_2) \land \mathit{subgoal}\,(g_2, g_3)) \Rightarrow \mathit{subgoal}\,(g_1, g_3)) \\
\forall g : \mathit{Goal} \bullet (\#\{g_1 : \mathit{Goal} \mid \mathit{subgoal}\,(g_1, g)\} \in \mathbb{N})
\end{array}
$$

Now, we say that a goal *benefits* another goal if the first is a subgoal of the second. This is formally described below.

$$
\begin{array}{|l}
\mathit{benefits}\_ : \mathbb{P}(\mathit{Goal} \times \mathit{Goal}) \\
\hline
\forall g_1, g_2 : \mathit{Goal} \bullet \\
\quad \mathit{benefits}\,(g_1, g_2) \Leftrightarrow \mathit{subgoal}\,(g_1, g_2)
\end{array}
$$

A goal *hinders* another if the first conflicts with one of the subgoals of the second. The formal representation of the *hinders* relation is given below. Notice that although a goal is a hindrance to another, it does not mean that this latter goal cannot be satisfied because agents can find other ways to satisfy their goals.

$$
\begin{array}{|l}
\mathit{hinders}\_ : \mathbb{P}(\mathit{Goal} \times \mathit{Goal}) \\
\hline
\forall g_1, g_2 : \mathit{Goal} \bullet \\
\quad \mathit{hinders}\,(g_1, g_2) \Leftrightarrow (\exists g_3 : \mathit{Goal} \bullet \\
\qquad (\mathit{subgoal}\,(g_3, g_2) \land \mathit{conflicts}\,(g_1, g_3)))
\end{array}
$$

3.4.5 Plans

Recipes for action that describe how a goal can be achieved are known as *plans*. They are usually described as sequences of actions that can be executed when certain conditions in the environment are satisfied. Since environments are dynamic, agents cannot know in advance how the world might change, nor opportunities or difficulties that they could face in the future. As a result, not all the details of a plan can be specified from the beginning [140]. Instead, some subgoals are included in plans to represent desired states, but without a corresponding subplan to achieve them. The selection of a plan for each subgoal is made only at the time at which the subgoal must be satisfied. The structure for plans adopted in this work is quite similar to that used in dMARS and AgentSpeak(L) [58, 59, 142]. First, we define a branch, or step, in a plan as either an action directly executed by an agent, or a goal (subgoal) that must be satisfied for the plan to continue.

$$Branch ::= actionstep\langle\langle Action\rangle\rangle \mid goalstep\langle\langle Goal\rangle\rangle$$

At execution time, when a subgoal in a plan is reached, a new plan is selected in order to satisfy that subgoal. In this way, the original plan is expanded to create a stack of plans as shown in Figure 3.2. The plan at the top of the stack corresponds to the most recent subgoal, and the plan at the base of the stack corresponds to the original goal. In the illustration, the original goal is g_0. A plan starting with an action a_1, a subgoal g_1, and an action a_2, is selected to satisfy it. After action a_1 is executed, a plan to satisfy g_1 is added to the top of the stack of plans. Then, actions a_3 and a_4 are executed and a plan to satisfy g_2 is added to the stack. As soon as the plan to satisfy g_2 finishes, it is removed from the top of the stack, and the plan to satisfy g_1 continues. When the stack is empty, the original goal could be considered as satisfied.

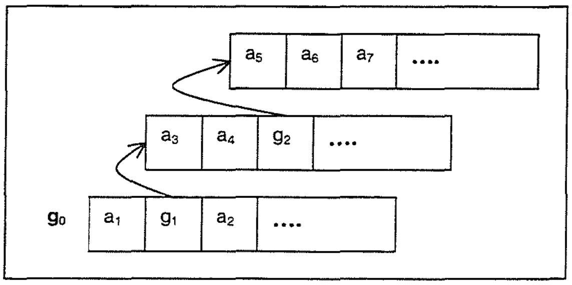


FIGURE 3.2: Stack of Plans

The *body* of a plan is a non-empty sequence of actions or goals. The most simple plan can be described as one whose body contains just an action. That is, the execution of the action leads to the satisfaction of the goal.

$$Body == \operatorname{seq} Branch$$

A general model of a plan is given in the schema below. It includes the *goal* that can be satisfied by executing all the actions and satisfying all the subgoals included in the *body*. The *context* is the state in the environment that must be true for a plan to be applied.

$$
\begin{array}{|l}
\hline
\textit{Plan} \\
\textit{goal} : \textit{Goal} \\
\textit{body} : \textit{Body} \\
\textit{context} : \mathbb{P}\,\textit{Attribute} \\
\hline
\end{array}
$$

In addition, functions to find either all actions (*planactions*) or subgoals (*plangoals*) included in a plan's body are defined as follows.

$$
\begin{array}{|l}
\textit{plangoals} : \textit{Body} \nrightarrow \mathbb{P}\,\textit{Goal} \\
\textit{planactions} : \textit{Body} \nrightarrow \mathbb{P}\,\textit{Action} \\
\hline
\forall\, b : \textit{Body};\ gs : \mathbb{P}\,\textit{Goal};\ \textit{acts} : \mathbb{P}\,\textit{Action} \bullet \\
\quad (\textit{plangoals}\ \ b = gs \Leftrightarrow (\forall\, g : gs \bullet \\
\qquad \exists\, br : \textit{Branch} \bullet br \in \operatorname{ran} b \land g = \textit{goalstep}^{-1} br)) \land \\
\quad (\textit{planactions}\ \ b = \textit{acts} \Leftrightarrow (\forall\, ac : \textit{acts} \bullet \\
\qquad \exists\, br : \textit{Branch} \bullet br \in \operatorname{ran} b \land ac = \textit{actionstep}^{-1} br))
\end{array}
$$

3.4.6 Intentions

Once a plan is selected to satisfy a goal, a *plan instance* is created. A plan instance is a *copy* of the original plan that now serves as a *mental attitude* directing behaviour as opposed to a *recipe* for behaviour. The distinction between plans as recipes and plans as mental attitudes is very important in the study of deliberative agents, and we distinguish them by calling the former *plans*, and the latter *plan instances*, whose formal representation is as follows.

$$\textit{PlanInstance} == \textit{Plan}$$

Once a goal is selected from an agent's desires, and a plan is selected to achieve it, the plan forms the basis of an *intention* that will direct the future behaviour of the agent [16]. An intention represents the commitments that one agent creates in order to achieve a goal [34]. As mentioned before, for each goal a sequence (or stack) of plan instances is created and all these plan instances are part of an intention. An intention is formally defined as a sequence of plan instances, and is represented as follows.

$$\textit{Intention} == \operatorname{seq} \textit{PlanInstance}$$

Defining intentions in this way gives agents flexibility for the achievement of goals because if an instantiated plan fails, agents have the opportunity to find another plan.

3.4.7 Agent Definition

To distinguish one agent from another, a unique name is assigned to each agent. The set of all agent names is defined below.

$$[\mathit{AgentName}]$$

An agent is an entity capable of satisfying some goals [62]. It has an identity (*self*) that makes it different from other agents. An agent is essentially defined by its *plan library*, which contains all the *recipes* for action the agent knows about, and its *capabilities* or specific actions. At run-time, an agent will typically have sets of *beliefs*, *intentions* and *goals* which are generated in response to changes in the environment through the reasoning and action control cycle of the agent. These components define the agent as it is acting in the world, and they are the key artifacts that are manipulated to ensure effective behaviour. The schema below formalises an agent.

$$
\begin{array}{|l}
\multicolumn{1}{l}{\mathit{Agent}} \\
\hline
\mathit{self} : \mathit{AgentName} \\
\mathit{planlibrary} : \mathbb{P}\,\mathit{Plan} \\
\mathit{capabilities} : \mathbb{P}_1\,\mathit{Action} \\
\mathit{beliefs} : \mathbb{P}_1\,\mathit{Belief} \\
\mathit{goals} : \mathbb{P}_1\,\mathit{Goal} \\
\mathit{intentions} : \mathbb{P}\,\mathit{Intention} \\
\hline
\mathit{planlibrary} \neq \varnothing \\
\mathit{capabilities} \neq \varnothing \\
\mathit{goals} \neq \varnothing \\
\hline
\end{array}
$$

3.5 Autonomous Agents

3.5.1 Motivations

According to Luck and d'Inverno [117, 120] *motivations* are any desires or preferences that can lead to the generation and adoption of goals, and which affect the outcome of any reasoning process intended to satisfy these goals. To represent motivations, first a set of symbols representing the identity of all motivations is defined.

$$[\mathit{MotiveSym}]$$

Each motivation has two associated elements: a *symbol* and an *intensity* [119]. The symbol is the identity of the motivation. The intensity is a value which represents how much an agent is motivated. This value changes according to an agent's beliefs, so that an agent's motivations are not always at the same level and, consequently, the focus of attention of an agent might change. The higher the intensity, the more motivated an agent. A schema for motivations is given as follows.

Motivation

$\mathit{symbol} : \mathit{MotiveSym}$
$\mathit{intensity} : \mathbb{N}$

3.5.2 Motivated Goals

Contrary to definitions that take motivations as goals [158, 159], the SMART framework clearly states the difference between them. Whereas goals are states that an agent wants to bring about, motivations are preferences that drive the behaviour of agents. Agents work to satisfy their goals, but when decisions must be taken, agent preferences are considered. The range of these decisions covers many aspects such as which goals to pursue, which goals to prefer, which goals to adopt, or even which society an agent wants to belong to. As Luck and d'Inverno state [117, 120], motivations are the main characteristic of autonomous agents.

In general, an autonomous agent's goals are associated with a unique set of motivations which are different for each agent. Thus, agents show their individual preferences towards particular goals. Then, it is said that an autonomous agent's goals are *motivated.* A *motivation-goal association* is formally defined as a relationship between a set of motivations and a goal, and is represented as follows.

$$\mathit{MotivationGoal} == \mathbb{P}\,\mathit{Motivation} \times \mathit{Goal}$$

The relationship above allows us to define the *importance* of a goal as the intensity of the highest of its associated motivations. The higher the motivation, the more important the goal. There might be other forms to define the importance of a goal. For instance, we can define it as the sum of all the intensities of the motivations associated with the goal or as the average of these intensities. For the purpose of this thesis it is not relevant which definition is chosen, only a means to express the preferences of an agent for a goal, and the means to compare two goals are needed.

A formal representation of a function to get the importance of a goal is given through the *goalimportance* axiomatic definition, which takes a set of motivation-goal associations and a goal as arguments. The function is divided into two cases. First, the importance of a goal is nil when there are no motivation-goal associations (i.e. agents are not autonomous) or there is no motivation-goal association corresponding to the required goal (i.e. the goal is not motivated). Otherwise, the importance of a goal is given by the motivation with the highest intensity.

$$
\begin{array}{|l}
goalimportance : (\mathbb{P}\, MotivationGoal \times Goal) \rightarrow \mathbb{N} \\
\hline
\forall gms : \mathbb{P}\, MotivationGoal;\ g : Goal;\ imp : \mathbb{N} \bullet \\
\quad (goalimportance\,(gms, g) = 0 \Leftrightarrow (gms = \varnothing \lor \\
\qquad \neg\,(\exists gm : gms \bullet g = second\ gm))) \lor \\
\quad (goalimportance\,(gms, g) = imp \Leftrightarrow (gms \neq \varnothing \land \\
\qquad (\exists gm : gms \mid g = second\ gm \bullet \\
\qquad\quad (\exists m : (first\ gm) \bullet (imp = m.intensity \land \\
\qquad\qquad (\forall m_2 : (first\ gm) \bullet imp \geq m_2.intensity))))))
\end{array}
$$

In Figure 3.3, an example of motivation-goal associations is shown. Vertical bars represent the intensity of the corresponding motivation of a particular goal. In the figure, goal g_1 is associated with three motivations, m_1, m_2, and m_3. The importance of g_1 is the intensity of motivation m_2. In the same way, goal g_2 is associated with two motivations, m_2 and m_4, and its importance is the intensity of the motivation m_4. Then, when comparing these goals, an agent prefers g_2 over g_1, because the importance of g_2 is higher than the importance of g_1.

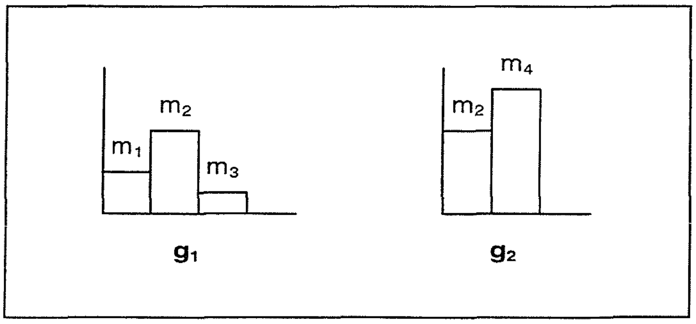


FIGURE 3.3: Motivated Goals

Instead of comparing two goals, sometimes comparing sets of goals is needed. Then, a way to define the *importance* of a set of goals must be given as well. As for the importance of a goal, there might be several alternatives to define the importance of a set of goals. For instance, we can either define it as the importance of the most motivated of the goals, or define it as the sum of the importance of each goal. In the first definition only one goal is considered to determine the importance of a complete set, whereas in the second, all the goals included in the set contribute to this value. Again, for our purposes, having the means to compare two goals and two sets of goals is enough, and only the first definition is considered. This is formally represented in the function below which states that the importance of a set of goals is defined as the importance of the most motivated goal in the set.

$$\begin{array}{|l}
\mathit{importance} : (\mathbb{P}\,\mathit{MotivationGoal} \times \mathbb{P}\,\mathit{Goal}) \rightarrow \mathbb{N} \\
\hline
\forall\, \mathit{gs} : \mathbb{P}\,\mathit{Goal};\ \mathit{gms} : \mathbb{P}\,\mathit{MotivationGoal};\ \mathit{imp} : \mathbb{N} \bullet \\
\quad \mathit{importance}\,(\mathit{gms}, \mathit{gs}) = \mathit{imp} \Leftrightarrow (\exists\, g_1 : \mathit{gs} \bullet \\
\qquad (\forall\, g_2 : \mathit{gs} \bullet \mathit{goalimportance}\,(\mathit{gms}, g_1) \geq \mathit{goalimportance}\,(\mathit{gms}, g_2)))
\end{array}$$

3.5.3 Autonomous Agent Definition

Based on the SMART framework, an *autonomous agent* is an agent with *motivations* which determine not only the goals that the agent is able to generate, but also its preferences. All of its goals are motivated (i.e. all goals have a unique set of associated motivations) and, therefore, the importance of each one of them can be obtained. The schema below represents an autonomous agent as an agent whose motivations are not empty (shown in the first predicate), and with a set of motivation-goal associations (*gms*). The second predicate states that for all goals, there exists a motivation-goal association with a set of motivations that is never empty. Finally, the last predicate states that only one set of motivations is associated with the same goal.

$$
\begin{array}{|l}
\textit{AutonomousAgent} \\
\hline
Agent \\
motivations : \mathbb{P}\, Motivation \\
gms : \mathbb{P}\, MotivationGoal \\
\hline
motivations \neq \varnothing \\
\forall g : goals \bullet (\exists_1\, gm : gms \bullet (first\, gm \neq \varnothing \wedge g = second\, gm)) \\
\forall ms : \mathbb{P}\, Motivation;\ g : goals;\ gm : gms \bullet \\
\quad (ms \subseteq motivations \wedge gm = (ms, g)) \Rightarrow \\
\qquad (\forall ms' : \mathbb{P}\, Motivation;\ gm' : gms \mid gm' = (ms', g) \bullet ms = ms') \\
\hline
\end{array}
$$

3.6 Conclusion

This chapter provides definitions for the elements considered necessary to develop our theory about norms, agents and multi-agent systems. The SMART agent framework has been fundamental for this labour. Particularly important is the notion of *motivated agency* which provides the basis for understanding the decisions that autonomous agents take. The range of these decisions covers many aspects such as which goals to pursue, which goals to prefer, which goals to adopt, or even which society an agent wants to belong to.

Besides providing definitions for classical concepts such as beliefs, goals, plans, intentions and motivations, the chapter provides formal ways to identify goals that are in conflict, goals that benefit from other goals, and goals that can be hindered by others. Identifying these relationships is important for agents to assess the consequences, on their goals, of satisfying external goals. Associations between goals and motivations to define *motivated goals* are also given in the chapter. By doing so, the *importance* of goals is defined. This is a key concept that will be used in the remains of this thesis to show how motivations drive the normative behaviour of agents. Thus, on the basis of the importance of goals many decisions regarding norms will be taken.

Agents and autonomous agents are defined following the SMART hierarchy of agents. However, these are not simple repetitions because we have extended their characteristics towards a model for normative agents. There are other aspects regarding motivations and goals that are not considered here, such as, how agents' motivations change according to changes in the environment, or how agents select goals to be intended based on their motivations. Although important, these aspects are not relevant for our work on norms but can be found elsewhere [83].

Chapter 4

A Normative Framework for Agent-Based Systems

4.1 Introduction

Since many conflicts of interest may appear when the actions of an agent negatively affect the goals of others, the behaviour of self-interested agents that coexist in a common world must be regulated [148]. The role of *norms* is precisely to avoid these conflicts because they prescribe what is permitted and what is forbidden in a society. Norms specify the responsibilities and benefits for the members of a society and, consequently, agents can make their plans for action based on the expected behaviour of others. Knowing what to expect from others may reduce the number of necessary interactions to achieve agreement among agents [3], so the complexity of some decision-making processes can also be reduced. Norms also formalise agreements between agents that promise to do something and agents that expect that thing to be done. In general, all kinds of activities that require the coordinated participation of more than one agent are possible thanks to the introduction of norms [94]. Given these characteristics, the introduction of norms in multi-agent systems has been considered as an important factor to increase the effectiveness of the work of agents [37, 45].

To incorporate norms in multi-agent systems, efforts have been done to describe and define the different types of norms that agents have to deal with [53, 157]. However, this work has not led towards a model that facilitates the computational representation of any kind of norm. Each kind of norm appears to be different, which also suggests that if we want to model agents able to reason about norms, different processes of reasoning must be proposed. There is also work that introduces norms in systems of agents to represent societies, institutions and organisations [51, 55, 69, 127, 144, 153]. This research has primarily been focused at the level of multi-agent systems where norms represent the means to achieve coordination among their members. There, agents are assumed to be able to comply with norms, to adopt new norms, and to obey the authorities of the system. Nothing is said about the reasons why agents will be willing to adopt and comply with norms, nor about how agents can identify situations in which an authority's orders are beyond its responsibilities. That is, although agents in such systems are said to be autonomous, their models of norms and systems regulated by norms do not offer the means to explain why *autonomous* agents that are working to satisfy their own goals, still comply with their social responsibilities.

We can say that there are two omissions in the introduction of norms into multi-agent systems. One is the lack of a canonical model of norms that facilitates their implementation, and that allows us to describe the processes of reasoning about norms. The other refers to considering, in the models of multi-agent systems regulated by norms, the perspective of individual agents and what they might need to effectively reason about the society in which they participate. Both are the concerns of this chapter, and the main objective is to present a formal framework for norms and normative multi-agent systems where emphasis is placed on those aspects that might affect an agent's goals and that can help agents in deciding what to do regarding norms.

The organisation of the chapter is as follows. Section 4.2 analyses different properties of norms. This analysis is then used to justify the elements that a general model of a norm must include in order to enable autonomous agents to reason about them. In Section 4.3 a discussion of different categories of norms is presented. These categories are formalised by using our proposed model of norms. In Section 4.4, the concepts of norm instances and interlocking norms are introduced, whereas in Section 4.5 the main properties of systems of autonomous agents that are regulated by norms are discussed, their components are defined, and a model is presented. This section also provides a way to identify general normative roles for agents. Section 4.6 analyses the dynamics of a system that results not only from the presence of norms, but also from the normative behaviour of agents within it, and defines the different possible states of a norm. Finally, a summary is given, the contributions are presented, and related work is compared and discussed.

4.2 Norms

4.2.1 Introduction

Norms are mechanisms to drive the behaviour of agents especially in those cases when their behaviour affects other agents. Norms can be characterised by their *prescriptive-*

ness, *sociality*, and *social pressure*. In other words,

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

By analysing these properties, the essential components of a norm can be identified. These components must enable agents to reason about why a norm should be complied with.

4.2.2 Norm Components

Norms specify patterns of behaviour for a set of agents. These patterns are sometimes represented as actions to be performed [5, 165], or restrictions to be imposed over an agent's actions [3, 133, 153]. At other times, patterns of behaviour are specified through goals that must be either satisfied or avoided by agents [39, 157]. Now, since actions are performed in order to change the state of an environment, goals are states that agents want to bring about, and restrictions can be seen as goals to be avoided, we argue that by considering goals the other two patterns of behaviour can be easily represented (as is shown later on in Section 4.2.4).

In brief, norms specify things that ought to be done and, consequently, a set of *normative goals* must be included in a norm. Sometimes, these normative goals must be directly intended, while at other times their role is to inhibit specific states (as in the case of prohibitions).

Norms are always directed at a set of *addressee agents* which are directly responsible for the satisfaction of the normative goals. The set of addressee agents may contain all the agents in the system, as with a mutually understood social law, or it might just contain a single agent. Moreover, sometimes to take decisions regarding norms, agents not only consider what must be done but also for whom it must be done, agents that *benefit* from the satisfaction of normative goals may also be included in a norm.

In general, norms are not applied all the time, but only in particular circumstances or within a specific *context*. Thus, norms must always specify the situations in which addressee agents must fulfill them. *Exception* states may also be included. These exception states represent situations in which addressees cannot be punished when they *have not* complied with norms. Exceptions represent *immunity* states for all addressee agents in a particular situation [146].

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

4.2.3 Norm Model

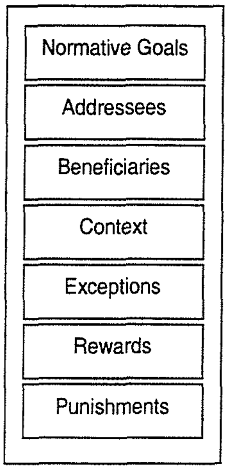


FIGURE 4.1: The Model of a Norm

Specifically, an agent may have access to certain norms which can be represented as data structures relating to social rules. Our proposed model of norms contains the components illustrated in Figure 4.1 and described as follows.

- A set of *normative goals* that the relevant group of agents must seek to achieve.
- Each norm applies to a certain set of agents, which may be all agents in a society, or just a limited subset of them. In either case, however, the *addressee* agents that should obey the norm are included.
- Typically, there is also a set of *beneficiary* agents, which are those agents that might specifically gain from addressee agents fulfilling the norm.

- The *context* of a norm refers to the enviromental state that must be believed by an agent for a norm to be complied with. For example, if an agent enters a library, the norm of being quiet must be triggered.

- The model also includes *exceptions*, which are states of the world that exempt addressee agents from the duties specified by the norm.

- Finally, it may be that addressee agents obtain some *reward* if norms are complied with, or *punishments* if they are not.

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

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

Norm

$normativegoals : \mathbb{P}\, Goal$

$addressees : \mathbb{P}\, AgentName$

$beneficiaries : \mathbb{P}\, AgentName$

$context : EnvState$

$exceptions : EnvState$

$rewards : \mathbb{P}\, Goal$

$punishments : \mathbb{P}\, Goal$

$normativegoals \neq \varnothing$

$addressees \neq \varnothing$

$context \neq \varnothing$

$context \cap exceptions = \varnothing$

$rewards \cap punishments = \varnothing$

4.2.4 Permitted and Forbidden Actions

Sometimes it is useful to observe norms not through the normative goals that ought to be achieved, but through the actions that can lead to the satisfaction of such goals. Then, actions that are either *permitted* or *forbidden* by a norm are considered as follows. If there is a situation state in which a norm must be fulfilled, and the results of an action benefit the achievement of the associated normative goal, then such an action is *permitted* by the respective norm. For example, the action of leaving a building through an emergency exit is an action that is permitted by the norm of being outside every time a fire alarm becomes activated. Formally, since both goals and the results of actions are defined in terms of states of the environment which are represented by a set of attributes, we say that an action is *permitted* by a norm in a particular state of the environment, if and only if the context in which such a norm must be applied is a subset of this state, and the results of the action benefit one of the normative goals of the norm (as defined in Subsection 3.4.4).

$$
\begin{array}{|l}
\mathit{permitted}\_ : \mathbb{P}(\mathit{Action} \times \mathit{Norm} \times \mathit{EnvState}) \\
\hline
\forall\, a : \mathit{Action};\; n : \mathit{Norm};\; \mathit{env} : \mathit{EnvState} \bullet \\
\quad \mathit{permitted}\ (a, n, \mathit{env}) \Leftrightarrow n.\mathit{context} \subseteq \mathit{env} \wedge \\
\qquad (\exists\, g : n.\mathit{normativegoals} \bullet \mathit{benefits}\ (a\ (\mathit{env}), g))
\end{array}
$$

By analogy, *forbidden* actions are defined as those actions leading to a situation which contradicts or hinders the normative goal. For example, the action *illegal parking* is an action forbidden by a norm whose normative goal is to avoid parking in front of a hospital entrance. Formally, we say that an action is *forbidden* by a norm in a particular state of the environment, if and only if the context in which such a norm must be applied is a subset of this state, and the results of the action hinder one of the normative goals of the norm. The definition of *hinders* predicate is given in Subsection 3.4.4.

$$
\begin{array}{|l}
\mathit{forbidden}\_ : \mathbb{P}(\mathit{Action} \times \mathit{Norm} \times \mathit{EnvState}) \\
\hline
\forall\, a : \mathit{Action};\; n : \mathit{Norm};\; \mathit{env} : \mathit{EnvState} \bullet \\
\quad \mathit{forbidden}\ (a, n, \mathit{env}) \Leftrightarrow n.\mathit{context} \subseteq \mathit{env} \wedge \\
\qquad (\exists\, g : n.\mathit{normativegoals} \bullet \mathit{hinders}\ (a\ (\mathit{env}), g))
\end{array}
$$

In other words, if an action is applied in the context of a norm, and the results of this action benefit the normative goals, then the action is permitted. However, when the action hinders the normative goals instead of providing benefits, then it is forbidden.

4.3 Categories of Norms

4.3.1 Introduction

The term *norm* has been used as a synonym for obligations [12, 54], prohibitions [52], social laws [153], and other kinds of rules imposed by societies (or by an authority). The position of our work is quite different. It considers that all these terms can be grouped in a general definition of a norm, because they have the same properties (i.e. prescriptiveness, sociality and social pressure) and they can be represented by using the same model. They all represent responsibilities for addressee agents, and create expectations for beneficiaries and other agents. They are also the means to support beneficiaries when they have to claim some compensation in the situations where norms are not fulfilled as expected. Moreover, whatever the kind of norm being considered, its fulfillment may be rewarded, and its violation may be penalised.

What makes one norm different from another is the way in which they are created, their persistence, and the components that are obligatory in the norm. Thus, norms might be created by an agent designer as built-in norms, they can be the result of agreements between agents, or they can be elaborated by a complex legal system. Regarding their persistence, norms might be taken into account during different periods of time, such as until an agent dies, as long as an agent stays in a society, or just for a short period of time until its normative goals become satisfied. Finally, some components of a norm might not exist; there are norms that include neither punishments nor rewards, even though they are complied with. Despite these differences, all types of norms can be reasoned about in similar ways. Some of these characteristics can be used to provide a *classification* of norms into four main categories: *obligations*, *prohibitions*, *social commitments* and *social codes* as shown in Figure 4.2. Below we explain each of these in turn.

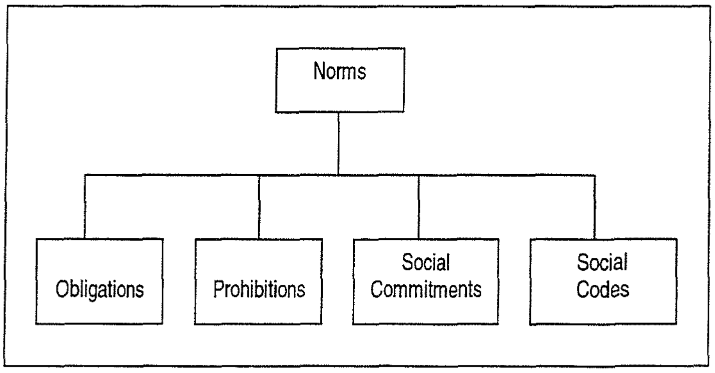


FIGURE 4.2: Categories of Norms

4.3.2 Obligations and Prohibitions

Obligations and *prohibitions* are norms whose purpose is to ensure the coordination of individuals in a society, and which agents adopt once they become members of the society. Agents adopt these norms because they represent the means to satisfy other important goals. Generally, addressee agents do not participate in their creation, but there are some agents entitled to do so. Obligations and prohibitions are considered by agents to be complied with, as long as they stay in a society. The main characteristic of these kinds of norms is that punishments are applied to those agents that offend them. Norms adopted by a secretary in an office, by workers in a factory, or by students in a university are some examples. Formally, an obligation is a norm in which violation is always penalised. To represent an obligation, the schema of a norm is used by imposing a constraint on punishments as follows.

$$\begin{array}{|l}
\textit{Obligation} \\
\hline
\textit{Norm} \\
\hline
\textit{punishments} \neq \varnothing \\
\hline
\end{array}$$

Whereas obligations represent goals that addressees must bring about, prohibitions represent goals that should be avoided. Since goals are represented as desired states, and states are represented as predicates or their negation, normative goals of prohibitions can be easily represented as negated goals. Consequently, no further distinction between obligations and prohibitions is given, and they have the same formal representation.

$\textit{Prohibition} == \textit{Obligation}$

4.3.3 Social Commitments

The second category of norms corresponds to *social commitments*. These are norms derived from agreements or negotiations between two or more agents [94]. They are part of a deal between two sets of agents and, consequently, addressees participate actively in their creation. Normative goals, rewards and punishments of this kind of norm are agreed rather than imposed. Once the normative goals of a social commitment are satisfied, rewards can be claimed. For this reason, social commitments sometimes come in pairs, one specifying what must be done in the first instance, and the other specifying what must be done when the first social commitment becomes fulfilled. Beneficiaries of a social commitment are, in general, responsible for monitoring its fulfillment. Contrary to obligations, social commitments are temporary, because they may disappear once the normative goals become satisfied. Social commitments are formally specified, in the schema below, as norms whose fulfillment is always rewarded.

SocialCommitment

Norm

$rewards \neq \varnothing$

4.3.4 Social Codes

Our third category of norms is *social codes.* These are norms that are accepted as general principles by the members of a society or a particular agent group. Rather than being forced through punishments or rewards, social codes are complied with as ends in themselves. They are motivated to be fulfilled because of the empathy or sympathy that addressee agents have towards other agents (specially towards agents that benefit from the norm), or because addressee agents want to express their social conformity. Examples of these kinds of norms can be norms that prescribe that elderly people must have priority for seats on buses, norms that state that garbage must not be thrown on the street, or norms that state that any personal information provided to an institution is confidential. Formally, social codes are norms which have neither punishments nor rewards (at least explicitly). They can be represented as follows.

SocialCode

Norm

$rewards = \varnothing$
$punishments = \varnothing$

In the remainder of this thesis, and in accordance with its definition, the term *norm* is used as an umbrella term to cover every type of norm, namely obligations, prohibitions, social commitments and social codes. The particular names will be referred to when needed.

4.3.5 Discussion

By using the proposed model, different kinds of norms varying from laws in a society, to norms in a family, obligations in an organisation, and even agreements among friends can be represented. Table 4.1 shows some raw examples of norms.

Social Law	Everyone must pay council tax during November, except full-time students, otherwise fines of £100 must be paid.
Family Rule	All children must be at home at 9:00 pm, otherwise they will not get dinner.
Job Regulation$_1$	All workers must produce n pieces of work during their working day, otherwise they will fired.
Job Regulation$_2$	All workers on the production line must receive a monthly payment as soon as they comply with *Job Regulation*$_1$.
Commitments	If Mike pays for the cinema tickets on Saturday, Ron will pay for dinner for both.

TABLE 4.1: Examples of Norms

NormativeGoals	Paying council tax
Addressees	All people over 18
Beneficiaries	City Council
Context	November each year
Exceptions	Full-time students
Rewards	–
Punishments	Fines up to £100

NormativeGoals	Being at home
Addressees	Children living in a house
Beneficiaries	–
Context	Every day at 9:00 pm
Exceptions	–
Rewards	–
Punishments	No dinner

TABLE 4.2: A Social Law and a Family Rule

The components of each norm of the Table 4.1 can be identified by making some assumptions. Table 4.2 shows, respectively, the representations of the social law, and the family rule. In both cases, normative goals, addressee agents, the context, states of exception, and punishments are easily identified, whereas rewards are not specified. Observe that the rule in a family represents the prohibition of being outside a house after 9 pm for all children living there.

NormativeGoals	Getting n pieces of work
Addressees	All workers
Beneficiaries	The company
Context	Every day
Exceptions	–
Rewards	Getting a salary
Punishments	Getting fired

NormativeGoals	Paying a salary
Addressees	Manager
Beneficiaries	A worker
Context	*Regulation*$_1$ is fulfilled
Exceptions	–
Rewards	–
Punishments	–

TABLE 4.3: Regulations in a Job

Table 4.3 shows the components of the norms in a factory. These norms are complementary because as soon as the first becomes fulfilled, the second must be considered to be fulfilled by the corresponding addressee agents. In the next section, these kinds of norms will be analysed because their structure allows the definition of interesting chains of norms. Finally, Table 4.4 shows the commitment of Table 4.1 between two friends expressed as two norms. That is, it is expected that once Mike fulfills his commitment of paying for cinema tickets on Saturday, he must receive, as a reward, a free dinner at Ron's expense. Once Ron receives the benefit of getting a free ticket for the cinema, he becomes committed to pay for the dinner for Mike. There are no associated punishments in both cases.

NormativeGoals	Pay for cinema tickets
Addressees	Mike
Beneficiaries	Ron
Context	On Saturday
Exceptions	Being ill
Rewards	Get a free dinner
Punishments	–

NormativeGoals	Pay for dinner
Addressees	Ron
Beneficiaries	Mike
Context	On Saturday after Mike pays the cinema
Exceptions	Ron has no money
Rewards	–
Punishments	–

TABLE 4.4: Commitments among Friends

4.4 Chains of Norms

4.4.1 Norm Instances

To understand the consequences of norms in a particular system, it is necessary to consider norms that are either fulfilled or unfulfilled. However, since most of the time a norm has a set of agents as addressees, the meaning of fulfilling a norm might depend on the interpretation of analysers of a system. In small groups of agents, it might be easy to consider a norm as fulfilled when every addressee agent has fulfilled the norm; by contrast, in larger societies, a proportion of agents complying with a norm will be enough to consider it as fulfilled. Instead of defining fulfilled norms in general, it is more appropriate to define norms being fulfilled by a particular addressee agent. To do so, the concept of norm instances is introduced.

Once a norm is adopted by an agent, a *norm instance* is created, which represents the internalisation of a norm by an agent. A norm instance is a copy of the original norm that is now used as a *mental attitude* from which new goals for the agent might be inferred. Norms and norm instances are the same concept used for different purposes. Norms are abstract specifications that exist in a society and are known by all agents [164], but agents work with *instances* of these norms. Consequently, there must be a separate instance for each addressee of a norm. Formally, we do not make any distinction between a norm and its instances, and an instance of a norm is represented as follows.

$$NormInstance == Norm$$

We say that a norm has been *fulfilled* by an addressee agent if all the normative goals of the corresponding instance have already been satisfied in a specific state. As can be observed, saying that an instance of a norm has been fulfilled is equivalent to saying that its normative goals have been satisfied. In what follows, we use both concepts without distinction. Formally, we say that an instance of a norm is fulfilled when all its normative goals are satisfied. Its formal representation is given in the schema below.

$$\begin{array}{|l}
\mathit{fulfilled}\_ : \mathbb{P}(NormInstance \times EnvState) \\
\hline
\forall n : NormInstance;\ st : EnvState \bullet \\
\quad \mathit{fulfilled}\ (n, st) \Leftrightarrow (\forall g : n.normativegoals \bullet \mathit{satisfied}\ (g, st))
\end{array}$$

Sometimes, it is important to know if an instance corresponds to a specific norm. Formally, we say that a norm instance corresponds to a norm if the addressee of the norm instance is an addressee of the norm, and each component of the norm instance corresponds to its counterpart in the norm. This is represented as follows.

$$\begin{array}{|l}
isnorminstance\_ : \mathbb{P}(NormInstance \times Norm) \\
\hline
\forall ni : NormInstance;\ n : Norm \bullet \\
\quad isnorminstance\ (ni, n) \Leftrightarrow \\
\qquad \#ni.addressees = 1 \land \\
\qquad ni.addressees \subseteq n.addressees \land \\
\qquad ni.normativegoals = n.normativegoals \land \\
\qquad ni.beneficiaries = n.beneficiaries \land \\
\qquad ni.context = n.context \land \\
\qquad ni.exceptions = n.exceptions \land \\
\qquad ni.rewards = n.rewards \land \\
\qquad ni.punishments = n.punishments
\end{array}$$

4.4.2 Interlocking Norms

The norms of a system are not isolated from each other; sometimes, compliance with them is a condition to trigger (or activate) other norms. That is, there are norms that prescribe how some agents must behave in situations in which other agents either comply with a norm or do not comply with it [146]. For example, when employees comply with their obligations in an office, paying their salary becomes an obligation of the employer; or when a plane cannot take-off, providing accommodation to passengers becomes a responsibility of the airline. Norms related in this way can make a complete chain of norms because the newly activated norms can, in turn, activate new ones. Now, since triggering a norm depends on past compliance with another norm, we call these kinds of norms *interlocking norms*. The norm that gives rise to another norm is called the *primary* norm, whereas the norm activated as a result of either the fulfillment or violation of the first is called the *secondary* norm.

In terms of the norm model mentioned earlier, the *context* is a state that must hold for a norm to be complied with. Since the fulfillment of a norm is assessed through its normative goals, the context of the secondary norm must include the satisfaction (or non-satisfaction) of all the primary norm's normative goals. Figure 4.3 illustrates the structure of both the primary and the secondary norms and how they are interlocked through the primary norm's normative goals and the secondary norm's context.

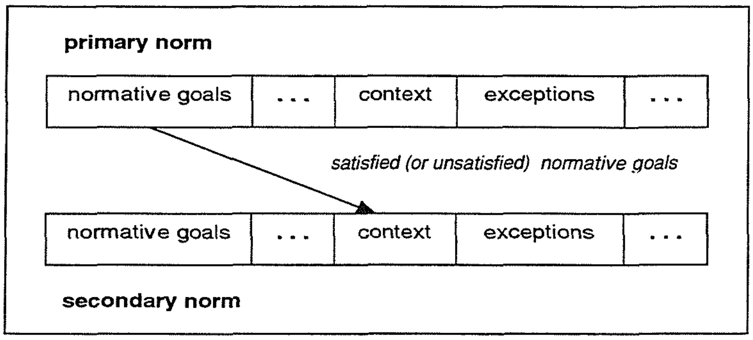


FIGURE 4.3: Interlocking Norm Structure

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

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

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

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

Having the means to relate norms in this way allows us to model how the normative behaviour of agents that are addressees of a secondary norm is influenced by the normative behaviour of addressees of a primary norm.

4.5 Normative Multi-Agent Systems

4.5.1 Introduction

Since norms are social concepts, they cannot be studied independently of the systems for which they are created and, consequently, an analysis of the normative aspects of social systems must be provided. Although social systems that are regulated by norms are different from one another, some general characteristics can be identified. They consist of a set of agents that are controlled by the same set of norms ranging from obligations and social commitments to social codes. However, whereas there are static systems in which all norms are defined in advance and agents in the system always comply with them [13, 153], a more realistic view of these kinds of systems suggests that when *autonomous* agents are considered, neither can all norms be known in advance (since new conflicts among agents may emerge and, therefore, new norms may be needed), nor can compliance with norms be guaranteed (since agents can decide not to comply). We can say then, that systems regulated by norms must include mechanisms to deal with both the modification of norms and the unpredictable normative behaviour of autonomous agents. In what follows, any kind of system of autonomous agents regulated by norms is called a *normative multi-agent system.* These systems have the following characteristics.

- *Membership.* Agents in a society must be able to deal with norms but, above all, they must recognise themselves as part of the system. This kind of social identification means that agents adopt the society norms and, by doing so, they show their willingness to comply with these norms.

- *Social Pressure*. Effective authority cannot be exerted if penalties or incentives are not applied when norms are either violated or complied with. However, this control must not be an agent's arbitrary decision, and although it is only exerted by some agents, it must be socially accepted.
- *Dynamism*. Normative systems are *dynamic* by nature. New norms are created and obsolete norms are abolished. Compliance or non-compliance with norms may activate other norms and, therefore, force other agents to act. Agents can either join or leave the system. The normative behaviour of agent members might be unexpected, and it may influence the behaviour of other agents.

Given these characteristics, we argue that normative multi-agent systems must include mechanisms to defend norms, to allow their modification, and to identify authorities. Their members must also be agents able to deal with norms. Each of these concepts is discussed in this section.

4.5.2 Enforcement and Reward Norms

Particularly interesting for this work are the norms triggered in order to punish offenders of other norms. We call them *enforcement norms* and their addressees are the *defenders* of a norm. These norms represent exerted social pressure because they specify not only who must apply the punishments, but also under which circumstances these punishments must be applied [146]. That is, once the violation of a norm becomes identified by defenders, their duty is to start a process in which offender agents can be punished. For example, if there is an obligation to pay accommodation fees for all students in a university, there must also be a norm stating what hall managers must do when a student refuses to pay.

As can be seen, norms that enforce other norms are a special case of interlocking norms because besides being interlocked by non-compliance, the normative goals of the secondary norm must include every punishment of the primary norm. Figure 4.4 shows how the structures of both norms are related. By modelling enforcement norms in this way, we cause an offender's punishments to be consistent with a defender's responsibilities. Addressees of an *enforced* norm (i.e. the primary norm) know what could happen if the norm is not complied with, and addressees of an *enforcement* norm (i.e. the secondary norm) know what must be done in order to punish the offenders of another norm. Enforcement norms allow the authority of defenders to be clearly constrained.

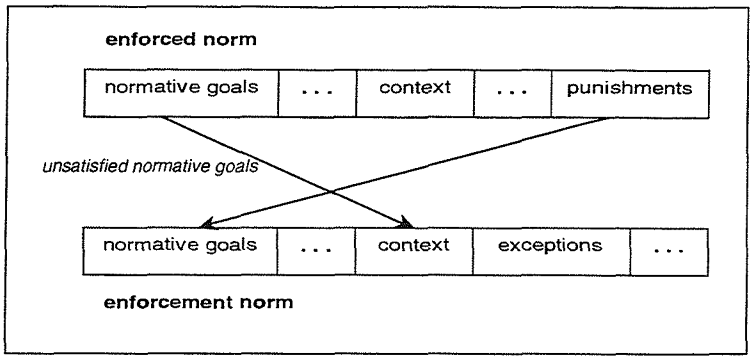


FIGURE 4.4: Enforcement Norm Structure

Formally, the relationship between a norm directed to control the behaviour of some agents and a norm directed at punishing the offenders of such a norm can be defined as follows. A norm *enforces* another norm if the first norm is activated when the second is violated, and all punishments associated with the violated norm are part of the normative goals of the first. Every norm satisfying this property is known as an *enforcement* norm.

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

So far we have described some interlocking norms in terms of punishments because punishments are one of the more commonly used mechanisms to enforce compliance with norms. However, a similar analysis can be done for interlocking norms corresponding to the process of rewarding members doing their duties. These norms must be interlocked by compliance and all the rewards included in the primary norm (rewarded norm) must be included in the normative goals of the secondary norm (reward norm). The relations between these norms are shown in Figure 4.5.

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

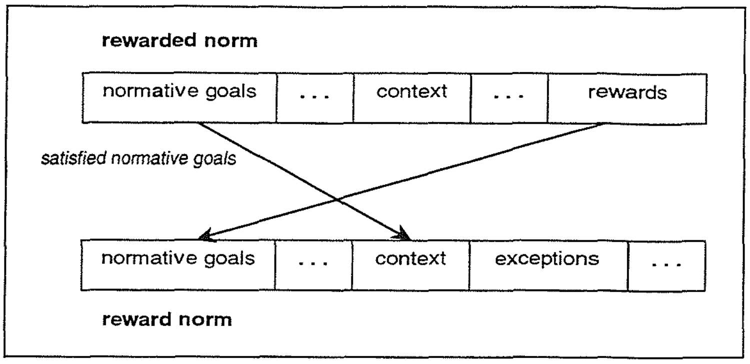


FIGURE 4.5: Reward Norm Structure

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

It is important to mention that this way of representing enforcement and reward norms can create an infinite chain of norms because we would also have to define norms to apply when authorities or defenders do not comply with their obligations either to punish those agents breaking rules or to reward those agents that fulfill their responsibilities [146]. The decision of when to stop this interlocking of norms is left to the creator of norms. If a system requires it, the model (and formalisation) for enforcing and encouraging norms can be used recursively as necessary. There is nothing in the definition of the model itself to prevent this.

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

4.5.3 Legislation Norms

Norms are introduced into a society as a means to achieve social order. Some are intended to avoid conflicts between agents, others to allow the establishment of commitments, and others still to unify the behaviour of agents as a means of social identification. However, neither all conflicts nor all commitments can be anticipated. Consequently, there must exist the possibility of creating new norms (to solve unexpected and recurrent conflicts among agents), modifying existing ones (to increase their effectiveness), or even abolishing those that become obsolete. Although it is possible that many of the members of a society have capabilities to do this, these capabilities must be restricted to be carried out by a particular set of agents in order to avoid everyone imposing norms, otherwise conflicts of interest might emerge. That is, norms stating when actions to legislate are permitted must exist in a normative multi-agent system [102]. Formally, we say that a norm is a *legislation* norm if actions to issue and to abolish norms are permitted by this norm in the current environment. These constraints are specified in the following declaration.

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

4.5.4 Normative Agents

The effectiveness of every structure of control relies on the capabilities of its members to recognise and follow its norms. However, given that agents are autonomous, the fulfillment of norms can never be taken for granted, since autonomous agents decide whether to comply with norms [116].

A *normative agent* is an agent whose behaviour is shaped by obligations that it has to comply with, prohibitions that limit the kind of goals that it can pursue, social commitments that are created during its social interactions, and social codes whose fulfillment represents social satisfaction for the agent, even though they are not penalised. Normative agents are able to deal with norms because they can represent, adopt, and comply with them and, for autonomous agents, decisions to adopt or comply with norms are made on the basis of their own goals and motivations. That is, autonomous agents are not only able to *act on* norms but also they are able to *reason about* them. In what follows, all normative agents are considered as autonomous agents that have adopted some norms and, although their normative behaviour is described in subsequent chapters, their representation is given in the schema below.

$$\begin{array}{|l}
\hline
\mathit{NormativeAgent} \\
\mathit{AutonomousAgent} \\
\mathit{norms} : \mathbb{P}\,\mathit{NormInstance} \\
\hline
\mathit{norms} \neq \varnothing \\
\hline
\end{array}$$

To remove any ambiguity in subsequent definitions, we assume that each normative agent in the world has a unique name, and that every agent name is associated with a unique normative agent. Formally, the *AgentWorld* schema is introduced. In this schema, the set of all agents in the world is represented by the variable *agents*, whereas *idagents* represents the set of all agent names. The two predicates in the schema state that each normative agent is associated with a unique agent name and that each agent name is associated with a unique normative agent, respectively.

$$\begin{array}{|l}
\hline
\mathit{AgentWorld} \\
\mathit{agents} : \mathbb{P}\,\mathit{NormativeAgent} \\
\mathit{idagents} : \mathbb{P}\,\mathit{AgentName} \\
\hline
\forall\, nag_1, nag_2 : \mathit{idagents};\ ag_1 : \mathit{agents} \bullet \\
\quad (ag_1.\mathit{self} = nag_1 \land ag_1.\mathit{self} = nag_2) \Rightarrow nag_1 = nag_2 \\
\forall\, nag : \mathit{idagents};\ ag_1, ag_2 : \mathit{agents} \bullet \\
\quad (ag_1.\mathit{self} = nag \land ag_2.\mathit{self} = nag) \Rightarrow ag_1 = ag_2 \\
\hline
\end{array}$$

A function (*normativeAg*) which, given an agent name, provides its corresponding normative agent model, is now specified as follows.

$$\begin{array}{|l}
\mathit{normativeAg} : \mathit{AgentName} \rightarrowtail \mathit{NormativeAgent} \\
\hline
\forall\, nag : \mathit{AgentName};\ ag : \mathit{NormativeAgent} \bullet \\
\quad \mathit{normativeAg}\,(nag) = ag \Leftrightarrow (\exists\, agW : \mathit{AgentWorld} \bullet \\
\qquad (nag \in agW.\mathit{idagents} \land ag \in agW.\mathit{agents} \land \\
\qquad ag.\mathit{self} = nag))
\end{array}$$

4.5.5 Normative Multi-Agent Systems Model

Having defined the components of a normative multi-agent system (*NMAS*), illustrated in Figure 4.6, a model of these kinds of systems can be provided. A normative multi-agent system includes a set of normative agents, called agent members, and a set of general norms that govern all of them. Subsets of these norms are dedicated to legisla-

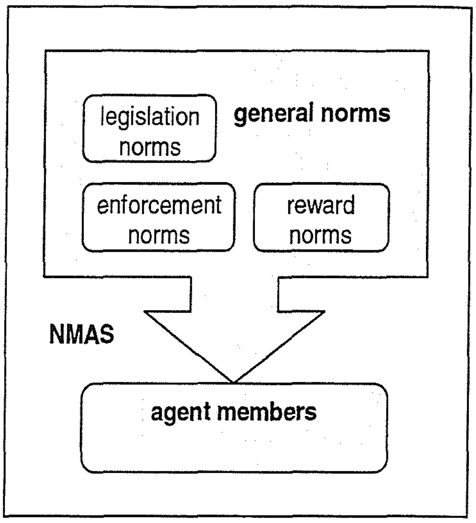


FIGURE 4.6: Normative Multi-Agent System Components

tion, others to punishing non-compliance with norms, and others to rewarding compliance with them. Now, since normative agents can belong to more than one normative multi-agent system, it is important to provide the means to distinguish one system from another. So, we introduce the set of names for all normative multi-agent systems as follows.

[*NMASName*]

A normative multi-agent system is formally represented in the *NormativeMAS* schema. It is defined in a world of agents, and it has an identity represented by the variable *nmasname*. A normative multi-agent system comprises a set of normative agent members (i.e. agents able to reason about norms) and a set of general norms that govern the behaviour of these agents (represented here by the variable *generalnorms*). There are also norms dedicated to enforcing other norms (*enforcenorms*), norms directed to encouraging compliance with norms through rewards (*rewardnorms*), and norms issued to allow the creation and abolition of norms (*legislationnorms*). The current state of the environment is represented by the variable *environment*. Constraints over these components are imposed as follows. The members of the system must be part of the world of agents (first predicate). Now, although it is possible that agents do not know all the norms in the system due to their own limitations, it is always expected that they at least adopt some norms, represented by the second predicate in the schema. The third predicate makes explicit that addressee agents of norms must be members of the system. Thus, addressee agents of every norm must be included in the set of member agents because it does not make any sense to have norms addressed to nonexistent agents.

The last three predicates respectively describe the structure of enforcement, reward and legislation norms. Notice that whereas every enforcement norm must have a norm to enforce, not every norm may have a corresponding enforcement norm, in which case no one in the society is legally entitled to punish an agent that does not fulfill such a norm.

$$
\begin{array}{|l}
\multicolumn{1}{l}{\_\_ \mathit{NormativeMAS} \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_} \\
\mathit{AgentWorld} \\
\mathit{nmasname} : \mathit{NMASName} \\
\mathit{members} : \mathbb{P}\,\mathit{AgentName} \\
\mathit{generalnorms} : \mathbb{P}\,\mathit{Norm} \\
\mathit{enforcenorms} : \mathbb{P}\,\mathit{Norm} \\
\mathit{rewardnorms} : \mathbb{P}\,\mathit{Norm} \\
\mathit{legislationnorms} : \mathbb{P}\,\mathit{Norm} \\
\mathit{environment} : \mathit{EnvState} \\
\hline
\mathit{members} \subset \mathit{idagents} \\
\forall\, \mathit{ag} : \mathit{members} \bullet (\mathit{normativeAg}\ \mathit{ag}).\mathit{norms} \cap \mathit{generalnorms} \neq \varnothing \\
\forall\, \mathit{sn} : \mathit{generalnorms} \bullet \mathit{sn}.\mathit{addressees} \subseteq \mathit{members} \\
\forall\, \mathit{en} : \mathit{enforcenorms} \bullet (\exists\, n : \mathit{generalnorms} \bullet \mathit{enforces}\ (\mathit{en}, n)) \\
\forall\, \mathit{rn} : \mathit{rewardnorms} \bullet (\exists\, n : \mathit{generalnorms} \bullet \mathit{rewardnorm}\ (\mathit{rn}, n)) \\
\forall\, \mathit{ln} : \mathit{legislationnorms} \bullet \mathit{legislate}\ (\mathit{ln}, \mathit{environment}) \\
\hline
\end{array}
$$

4.5.6 Normative Roles

Defining normative multi-agent systems in this way allows the identification of general roles for agents as follows. Besides roles of addressees and beneficiaries of a norm described earlier, there are other roles that depend on the kind of norms agents are responsible for. All possible roles are listed below.

- *Addressee* agents are directly responsible for the achievement of normative goals.
- *Beneficiaries* are agents whose goals can benefit from normative goals becoming satisfied.
- The set of agents that are entitled to create, modify, or abolish norms is called *legislators*. No other members of the society are endowed with this authority, and generally they are either elected or decreed by other agents.
- *Defender* agents are directly responsible for the application of punishments when norms are violated. That is, their main responsibility is to monitor compliance

with norms in order to detect transgressions. Moreover, they can also warn agents by advertising the bad consequences of being rebellious.

- By contrast, *promoter* agents are those whose responsibilities include rewarding compliant addressees. These agents also monitor compliance with norms in order to know when rewards must be given, and instead of *enforcing* compliance with norms they simply *encourage* it.

These *normative roles* for agents are not mutually exclusive. In fact, agents are able to have more than one normative role at the same time, depending on the kind of norm being considered. For example, in a social commitment, beneficiary agents can also be defenders and encourage the fulfillment of a norm. They can even apply sanctions or give the agreed rewards. In an office, the manager can be both a legislator and impose his own norms, and a defender entitled to punish his employees. The more complex a society, the more elaborate these normative roles become and, in some cases, legislators and defenders constitute a complex structure of control generally named *government*, with its own legal norms directed at managing the rest of the society.

Both addressees and beneficiaries can be directly observed in the structure of a norm. By contrast, legislators, defenders and promoters can only be observed within the context of a normative multi-agent system which gives them the scope of their entitlements (i.e. the authority of these agents is only recognised by members of the same system, no other agent ought to obey them). Formally, the *authorities* of a system are defined as the addressee agents of every legislation, enforcement or reward norm. They are represented in the schema below.

$$
\begin{array}{|l}
\hline
\mathit{AuthoritiesNMAS} \\
\mathit{NormativeMAS} \\
\mathit{legislators} : \mathbb{P}\,\mathit{AgentName} \\
\mathit{defenders} : \mathbb{P}\,\mathit{AgentName} \\
\mathit{promoters} : \mathbb{P}\,\mathit{AgentName} \\
\hline
\forall\, \mathit{lg} : \mathit{legislators} \bullet (\exists\, \mathit{lnorm} : \mathit{legislationnorms} \bullet \mathit{lg} \in \mathit{lnorm.addressees}) \\
\forall\, \mathit{df} : \mathit{defenders} \bullet (\exists\, \mathit{enorm} : \mathit{enforcenorms} \bullet \mathit{df} \in \mathit{enorm.addressees}) \\
\forall\, \mathit{pm} : \mathit{promoters} \bullet (\exists\, \mathit{rnorm} : \mathit{rewardnorms} \bullet \mathit{pm} \in \mathit{rnorm.addressees}) \\
\hline
\end{array}
$$

As can be seen, all components of a normative multi-agent system cannot be taken independently, but are somehow complementary.

4.6 Dynamics of Norms

4.6.1 Introduction

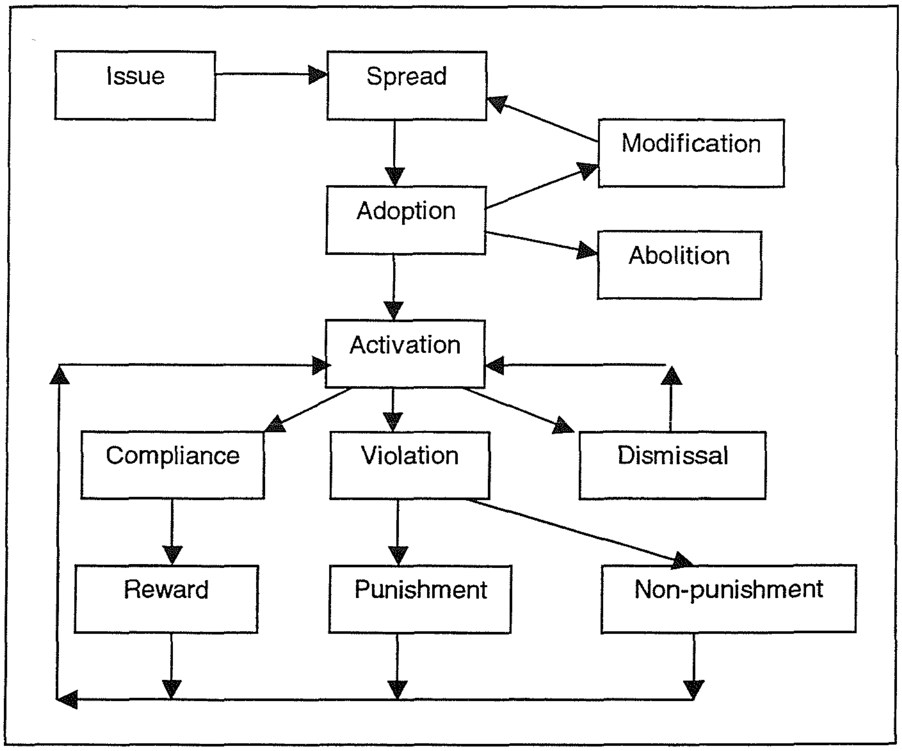


FIGURE 4.7: Norm Dynamics

Norms are not a static concept. Their inclusion in a system influences the behaviour of those agents responsible for complying with them, those agents that benefit from them, and those agents responsible for monitoring the normative behaviour of other agents. There are different processes started by norms (ranging from their creation to their abolition) in which different agents become involved. From these processes, the states of a norm can be identified. Figure 4.7 shows the transitions between one state of a norm to another as follows. First, legislators issue a norm. After that, the norm is spread among the agent members by either indirect or direct communication. Then, adoption of norms by addressee agents takes place, and instances of the norm are created; through this process an agent expresses its willingness to fulfill the norm as a way of being part of the society. Once a norm is adopted, it remains inactive, or in *latency*, until the context (which represent the applicability conditions) is satisfied. In exception states, agents are not obliged to comply with these norms, and consequently norms can be ignored. However, in most cases, two different situations might occur after a norm becomes activated, depending on whether the norm is fulfilled by addressee agents. After a norm is complied with, a reward can be offered. By contrast, if the norm is violated a punishment is applied. However, since agents responsible for the application of punishments have limited perception it is possible that the violation of a norm remains unnoticed and, therefore, offenders are not punished. Finally, as time progresses, some norms are either abolished or modified.

States of norms are the result of both the normative behaviour of different agents and changes in the environment. For instance, norms are issued by legislators but are adopted and complied with by addressees, and norms are activated when the environment state satisfies their context. Identifying the different states of norms is important because changes to them cause agents to react and, consequently, the way in which the normative behaviour of agents might be influenced by the normative behaviour of other agents can be modelled. For example, addressee agents acquire new responsibilities because of adopted norms, beneficiary agents might require compliance with active norms, and defender agents might apply punishments to the addressees of unfulfilled norms. In the following subsections, the way in which these states of norms are identified is explained.

4.6.2 Changing Norms

Legislation of norms is a responsibility only shouldered by legislator agents. Such a responsibility comprises at least three processes, namely: issuance, abolition, and modification of norms. These processes involve changes that might affect any agent in the system. Consequently, analysis of the prevailing situation and how the changes might affect the complete society are needed before any change can be made. Situations of this kind are complex and some of them have been investigated by researchers working on *emergence of norms* [5, 11, 87, 167, 171]. All these problems are beyond the scope of this thesis and, therefore, the processes to issue, abolish and modify norms are not provided, but the changes that result from any modification in the system of norms can be explained.

After a legislator decrees either the creation of a new norm or the modification or abolition of an old one, these events must be notified (spread) to all agents in the society. As a result of these changes at a global level, some of the agent members might also change because new norms might be adopted, and other norms might be modified or even abolished. Before explaining these changes, a relationship that holds between a norm and the legislator that issues it is formalised by using the predicate below.

$$\mathit{issuedby}\_ : \mathbb{P}(\mathit{Norm} \times \mathit{AgentName})$$

The *NormLegislation* schema formalises all the functions associated with the legislation of norms. Thus, the legislation of norms is defined in a normative multi-agent system where authorities can be identified. We represent this by including the *NormativeMAS* and *AuthoritiesNMAS* schemas. Two functions to identify all recently created norms (*getnewnorms*), and all norms that must be abolished (*getobsoletenorms*) are introduced as well. Notice that since the modification of norms can be seen as the abolition of a subset of norms together with the issuance of another subset of norms with the same name, a specific function to modify norms is not needed. The functions *spreadnorms* and *abolishnorms*, which can be seen as the processes through which agents are notified of the creation of new norms and the abolition of norms that become obsolete, are also included. The two predicates in the schema state that only legislators are entitled to create or abolish norms.

$$
\begin{array}{|l}
\multicolumn{1}{l}{\textit{NormLegislation}} \\ \hline
\textit{NormativeMAS} \\
\textit{AuthoritiesNMAS} \\
\textit{getnewnorms} : \mathbb{P}\,\textit{AgentName} \nrightarrow \mathbb{P}\,\textit{Norm} \\
\textit{getobsoletenorms} : \mathbb{P}\,\textit{AgentName} \nrightarrow \mathbb{P}\,\textit{Norm} \\
\textit{spreadnorms} : (\mathbb{P}\,\textit{AgentName} \times \mathbb{P}\,\textit{Norm}) \nrightarrow \mathbb{P}\,\textit{AgentName} \\
\textit{abolishnorms} : (\mathbb{P}\,\textit{AgentName} \times \mathbb{P}\,\textit{Norm}) \nrightarrow \mathbb{P}\,\textit{AgentName} \\ \hline
\forall\, nn : (\operatorname{ran} \textit{getnewnorms}) \bullet \\
\quad (\exists\, lag : \textit{legislators} \bullet (\forall\, n : nn \bullet \textit{issuedby}\ (n, lag))) \\
\forall\, on : (\operatorname{ran} \textit{getobsoletenorms}) \bullet \\
\quad (\exists\, lag : \textit{legislators} \bullet (\forall\, n : on \bullet \textit{issuedby}\ (n, lag))) \\ \hline
\end{array}
$$

In the *ChangeLegislation* schema, the operation for updating the norms in a system according to the changes dictated by legislators is specified as follows. First, all norms recently created (*newnorms*) and all norms that must be abolished (*obsoletenorms*) are obtained. After that, agents in the system are notified about which norms are obsolete and must, therefore, be removed. The variable *agentsabolish* represents the agents after the abolition of some of their norms. Then, these agents are updated with all recently created norms. Finally, the set of system norms is updated, and consists of all the old norms except those recently abolished, together with all norms recently created.

$$\begin{array}{l}
\textit{ChangeLegislation} \\
\hline
\Delta \textit{NormativeMAS} \\
\textit{NormLegislation} \\
\hline
\textbf{let } \textit{newnorms} == \textit{getnewnorms}\ (\textit{legislators}) \bullet \\
(\textbf{let } \textit{obsoletenorms} == \textit{getobsoletenorms}\ (\textit{legislators}) \bullet \\
(\textbf{let } \textit{agentsabolish} == \textit{abolishnorms}\ (\textit{members}, \textit{obsoletenorms}) \bullet \\
\quad (\textit{members}' = \textit{spreadnorms}\ (\textit{agentsabolish}, \textit{newnorms}) \land \\
\quad \textit{generalnorms}' = \textit{generalnorms} \setminus \textit{obsoletenorms} \cup \textit{newnorms})))
\end{array}$$

4.6.3 Norm States

Once norms are adopted, instances of norms are created by addressee agents. Remember that an instance of a norm is just a copy of the original norm which an addressee works with. At a very high level (i.e. from the perspective of an external observer), all instances of norms remain in a cycle until they become abolished. This cycle starts when a norm instance becomes activated. A norm instance is *active* when its context is satisfied in the current environmental state. For example, if a driver wants to park his car in front of an entrance, the norm that forbids such situations is applied, otherwise the norm is not even considered by the driver. Formally, we say that a norm instance is active when its context is a logical consequence (defined in Subsection 3.4.3) of the state of the environment. This is specified in the following predicate.

$$\begin{array}{l}
\textit{activenorm\_} : \mathbb{P}(\textit{NormInstance} \times \textit{EnvState}) \\
\hline
\forall\, n : \textit{NormInstance};\ st : \textit{EnvState} \bullet \\
\quad \textit{activenorm}\ (n, st) \Leftrightarrow \textit{logicalconsequence}\ (st, n.\textit{context})
\end{array}$$

The cycle of norm instances continues when these instances become either fulfilled or violated (as defined earlier). Fulfilled instances might provoke the activation of the corresponding norm to reward the compliant addressee. We say that a norm instance has been *rewarded* if it has been fulfilled and the corresponding norm to reward it has been also fulfilled. This means that the promoter of the norm has also complied with its responsibility of rewarding compliance with norms. Something similar occurs with unfulfilled norm instances, which might cause the activation of enforcement norms to punish the corresponding offender. We say that a norm instance has been *punished* if it has been violated and the corresponding enforcement norm has already been fulfilled. Here, the defender of the norm has complied with its obligation of punishing agents. Since norms and their corresponding enforcement and reward norms are defined in relation to a normative multi-agent system, to formalise them first we define the state of a system.

The *NMASState* schema represents the states of all the norm instances in a system. The variable *allinstances* represents the instances of each of the norms in the system, whereas *activenorms* variable represents the norm instances currently active. The schema also includes variables to represent norm instances that have been fulfilled (*fulfillednorms*), violated (*unfulfillednorms*), rewarded (*rewardednorms*), and punished (*punishednorms*). In the predicate part of the schema, the states of norm instances are defined as follows. The first predicate states that all norm instances are instances of a general norm. The next three predicates define active, fulfilled and unfulfilled norm instances as explained earlier. The fifth predicate states that for all rewarded norm instances, there must be an already fulfilled reward norm. The last predicate states that punished norm instances are those for which the corresponding enforcement norm has already been fulfilled. Notice that all norm states are taken according to the current environment of the system.

$$
\begin{array}{|l}
\multicolumn{1}{l}{\textit{NMASState}} \\
\hline
\textit{NormativeMAS} \\
\textit{allinstances} : \mathbb{P}\,\textit{NormInstance} \\
\textit{activenorms} : \mathbb{P}\,\textit{NormInstance} \\
\textit{fulfillednorms} : \mathbb{P}\,\textit{NormInstance} \\
\textit{unfulfillednorms} : \mathbb{P}\,\textit{NormInstance} \\
\textit{rewardednorms} : \mathbb{P}\,\textit{NormInstance} \\
\textit{punishednorms} : \mathbb{P}\,\textit{NormInstance} \\
\hline
\forall\, \textit{in} : \textit{allinstances} \bullet (\exists\, n : \textit{generalnorms} \bullet \textit{isnorminstance}\,(\textit{in}, n)) \\
\forall\, \textit{na} : \textit{activenorms} \bullet \textit{activenorm}\,(\textit{na}, \textit{environment}) \\
\forall\, \textit{fn} : \textit{fulfillednorms} \bullet \textit{fulfilled}\,(\textit{fn}, \textit{environment}) \\
\forall\, \textit{ufn} : \textit{unfulfillednorms} \bullet (\neg\, \textit{fulfilled}\,(\textit{ufn}, \textit{environment})) \\
\forall\, \textit{rn} : \textit{rewardednorms} \bullet (\exists\, \textit{rgn} : \textit{rewardnorms} \bullet \\
\quad (\textit{rewardnorm}\,(\textit{rgn}, \textit{rn}) \wedge \textit{fulfilled}\,(\textit{rgn}, \textit{environment}))) \\
\forall\, \textit{pn} : \textit{punishednorms} \bullet (\exists\, \textit{egn} : \textit{enforcenorms} \bullet \\
\quad (\textit{enforces}\,(\textit{egn}, \textit{pn}) \wedge \textit{fulfilled}\,(\textit{egn}, \textit{environment}))) \\
\hline
\end{array}
$$

This schema can be used by agents to assess the normative behaviour of other agents. For instance, an agent is an offender of a norm if the corresponding instance is an unfulfilled norm.

Now, although not all norm instances change their state at the same time, they must be updated at a particular point in time. At a particular time some instances of norms become activated, and other previously activated norm instances become either fulfilled or violated. Some of the unfulfilled norm instances are punished, and some of the fulfilled ones are rewarded. These changes are represented in the *UpdatingNormStates* operation schema. It includes the function (*observedchanges*) which reports the observed changes in the social environment. Then, the state of norm instances change as follows. First, the variable *environment* takes the new state of the environment. Next, sets of instances of norms are updated as follows. The set of new active norms (*newactive*) is calculated by analysing if the context, to trigger a norm, is true in the current state of the system. After that, the set of active norms that were fulfilled (*newfulfilled*) by their corresponding addressee agents is calculated by verifying the satisfaction of the corresponding normative goals. Next, unfulfilled norms that were punished (*newpunished*) are found by verifying if the norms that enforces them, have already been satisfied. Something similar is done to verify if fulfilled norms were rewarded (*newrewarded*). In this way, the states of norms are updated accordingly. These changes are represented in the last five lines of the predicate part of the schema as follows. Active norms (*activenorms*) are replaced by the set of new active norms, and the sets *fulfillednorms*, *unfulfillednorms*, *punishednorms* and *rewardednorms* are increased respectively by all the active norms already fulfilled, unfulfilled, unfulfilled norms that were punished, and fulfilled norms that were rewarded.

$$
\begin{array}{|l}
\mathit{UpdatingNormStates} \\
\hline
\Delta \mathit{NMASState} \\
\mathit{observedchanges} : \mathit{EnvState} \nrightarrow \mathit{EnvState} \\
\hline
\mathit{environment}' = \mathit{observedchanges}\ (\mathit{environment}) \\
(\mathbf{let}\ \mathit{newactive} == \{n : \mathit{allinstances} \mid \mathit{activenorm}\ (n, \mathit{environment}')\} \bullet \\
(\mathbf{let}\ \mathit{newfulfilled} == \{n : \mathit{activenorms} \mid \mathit{fulfilled}\ (n, \mathit{environment}')\} \bullet \\
(\mathbf{let}\ \mathit{newpunished} == \{n : \mathit{unfulfillednorms} \mid (\exists\, \mathit{en} : \mathit{enforcenorms} \bullet \\
\quad \mathit{fulfilled}\ (\mathit{en}, \mathit{environment}'))\} \bullet \\
(\mathbf{let}\ \mathit{newrewarded} == \{n : \mathit{fulfillednorms} \mid (\exists\, \mathit{en} : \mathit{enforcenorms} \bullet \\
\quad \mathit{fulfilled}\ (\mathit{en}, \mathit{environment}'))\} \bullet \\
(\mathit{activenorms}' = \mathit{newactive} \land \\
\mathit{fulfillednorms}' = \mathit{fulfillednorms} \cup \mathit{newfulfilled} \land \\
\mathit{unfulfillednorms}' = \mathit{unfulfillednorms} \cup (\mathit{activenorms} \setminus \mathit{newfulfilled}) \land \\
\mathit{punishednorms}' = \mathit{punishednorms} \cup \mathit{newpunished} \land \\
\mathit{rewardednorms}' = \mathit{rewardednorms} \cup \mathit{newrewarded})))))
\end{array}
$$

4.7 Conclusion

4.7.1 Summary

Norms are mechanisms to influence the social behaviour of agents. They have three properties: *prescriptiveness*, *sociality*, and *social pressure*. A norm prescribes how an agent must behave in situations in which more than one agent is involved. It is always expected that norms conflict with the individual goals of agents, so mechanisms to enforce compliance with them are always needed. In this chapter, a general model of norms has been proposed. It includes components that allow agents to know what must be done, when it must be done, who is responsible for what, who may benefit from this, and what may happen in cases in which norms become either fulfilled or violated. By considering these components in a model of norms, it is possible to represent different categories of norms ranging from *obligations* and *prohibitions*, to *social commitments* and *social codes*. Some cases of norms require that instead of stating what must be done, norms must prescribe what actions are permitted and what actions are forbidden for agents. However, instead of proposing a different model, these cases are considered as variants of the general model of norms.

Norms are social concepts and, therefore, they cannot be studied outside the system for which they are created. Norms form a complete system in which many norms are interlocked. Thus, compliance with some norms cause other norms to be activated. In this chapter, a model for such norms has been proposed. Normative multi-agent systems are defined as systems of autonomous agents controlled by a common set of norms. These systems have three characteristics: *membership*, *social pressure*, and *dynamism*. Since their members are autonomous normative agents able to reason about norms, compliance with norms cannot be guaranteed. We have considered this, and the proposed model of normative systems includes mechanisms to either enforce or promote the fulfillment of norms. The components of our model enable agents to recognise the system's authorities, and the limits of their authority.

Normative agents are defined here as autonomous agents able to reason about norms. These agents besides being able to represent norms, can adopt and comply with norms not as an end, but by considering their own goals and motivations to decide whether to do so. We argue that besides agent members and the general norms directed at them, normative multi-agent systems must include norms that allow agents to legislate, punish and reward compliance with norms. Through these norms, order in a system is reached because the authority and responsibilities of agents are well defined. Moreover, by using these norms, the roles of legislators, defenders and promoters of norms are easily identified.

Given the autonomy of their members, normative multi-agent systems are by nature *dynamic*. Such dynamism results from the agent decisions about whether to adopt or comply with norms because the normative behaviour of agents influences the normative behaviour of other agents. In other words, the fulfillment or violation of norms might cause other agents to act. Towards a modelling of the normative behaviour of agents, we have identified (and defined) different states for norm instances. At a determined time some norm instances are activated, others are either fulfilled or violated, the already fulfilled instances of norms become rewarded, and the violated ones become punished.

4.7.2 Contributions

A well defined framework for norms and normative multi-agent systems is the main contribution of this chapter. The framework contributes to a better understanding of the role of norms, because besides identifying the main properties of norms and normative multi-agent systems, models of them that can be computationally implemented have been provided. Moreover, the dynamism of normative multi-agent systems has been explained as a result of the normative behaviour of their members. States of norm instances have been discussed, and the means by which agents can identify them have been given. Although there has been much research on norms and agents, the work presented here is more complete because it not only subsumes most of the current work on norms, but it also covers many aspects not considered previously. Particular contributions of the chapter can be listed as follows.

- A general model of norms has been developed with the following characteristics.
    - It provides the means for autonomous agents to decide why and when norms should be complied with.
    - It can be used to group together different known categories of norms in a coherent fashion to provide a unifying model.
    - Its structure allows the representation of complex chains of norms.
- A general model of normative multi-agent systems has been developed with elements to deal with the autonomy of their members.
    - It includes legislation norms that allow the modification of norms. In this way, the norms of a system might change at any time without loosing the required control because not all agents are entitled to modify them.
    - Enforcement and reward norms are included as mechanisms to either enforce or promote compliance with norms.

  - The authority of agents is very well defined and constrained through legislation, reward and enforcement norms.

- A way to identify the different states of a norm has been elaborated by using norm instances.

In the remainder of this subsection, comparisons between some of the contributions of this work and the work developed by others are presented.

Table 4.5 compares the model for norms proposed in this thesis (denoted by *López*) against models proposed by others. This comparison is made on the basis of which components have been included in other models of norms. In particular, the following components (represented in columns) are assessed: the prescribed pattern of behaviour (*PB*); the addressees of the norm (*AD*); the conditions to trigger a norm(*CN*); the situations of exception (*EX*); the beneficiaries of the norm (*BN*); the incentives to comply with the norm (*RW*); and the punishments to avoid deviation from the norm (*PN*). Ticked columns mean that such a component has been included in the model, whereas a dash (−) means it is not included. Finally, *not explicit* means that although the component is mentioned by authors, it is not explicitly represented in their model. In this table not all perspectives discussed in Chapter 2 are included because many are quite similar; only the more representative models are considered. In particular, our model for norms is compared with the model of norms provided by Ross [146], Tuomela [165], Axelrod [5], Ullmann-Margalit [167], Conte and Castelfranchi [39, 41], Dignum [52], and Singh [157].

Norm Model	*PB*	*AD*	*CN*	*EX*	*BN*	*RW*	*PN*
Ross	✓	✓	✓	not explicit	✓	–	not explicit
Tuomela	✓	✓	✓	–	–	–	not explicit
Axelrod	✓	all members	–	–	–	not explicit	meta-norms
Ullmann-Margalit	✓	all members	–	–	–	–	not explicit
Conte & Castelfranchi	✓	✓	–	–	✓	–	not explicit
Dignum	✓	✓	–	–	✓	–	–
Singh	✓	✓	–	not explicit	✓	–	not explicit
López	✓	✓	✓	✓	✓	✓	✓

TABLE 4.5: Models of Norms

As can be observed from the table, the majority of the proposed models consider only three things when modelling norms: what must be done, who must do it, and for whom this must be done. We claim that for autonomous agents to deliberate about norms, more components must be considered in the representation of norms. Specifying the situations in which norms must be applied is important because they are not applied all the time but only in specific agent states. Immunity conditions give flexibility to the system by considering special situations in which agents can dismiss norms. Since agents are autonomous, they must know the consequences of complying with norms in order to motivate them to fulfill the norms. Consequently, norm models must include the rewards that can be obtained by complying with the norm and the punishments that might be applied if it is not. In Axelrod's model, metanorms are used to punish those offenders of norms. Our model allows the representation of these kinds of norms by using interlocking norms.

We can conclude that the model of norms presented in this chapter subsumes other models and is more appropriate for agents that do not comply with norms as an end, but that reason about why norms should be complied with. For instance, Esteva et al.'s model of norms [68, 69, 70] can be seen as an instance of our own model of norms in which the context of a norm is taken sets of messages that have and have not been issued, and a set of constraints that must be fulfilled. In this case, normative goals are the issuance of new messages. In addition, scenes of this particular framework can be defined as interlocking norms, because a norm is satisfied when the agent issues the corresponding messages which, in turn, activate another norm, causing the corresponding addressee agent to act accordingly.

Our model of norms differs from others [13, 127, 165] in the way in which patterns of behaviour are prescribed. To describe the pattern of behaviour prescribed by a norm, other models use actions. Thus, agents are told what exactly they must do. By contrast, we use normative goals, which is an idea more compatible with autonomous agents whose behaviour is driven by goals. Agents can choose the way to satisfy the normative goals, instead of being told exactly how it must be done. Our work also emphasises that all norms can be represented by using similar components, and that they are analysed by agents in similar ways. We also consider that what makes one norm different from another is the way norms are created, how long they are valid, and the reasons agents have to adopt them. These factors enable norms to be divided into categories such as obligations and prohibitions, social commitments and social codes.

A collateral result of our work is the proposed model for interlocking norms. These relations between norms have already been mentioned in several papers, especially from philosophical and legal perspectives [146], but no ways to model them have been provided. Dignum's concept of authorisations [53] attempts to describe norms activated when others are not fulfilled; however, his idea and models are incomplete. We claim that this form of representing connections between norms can be used not only to represent enforcement and reward norms, but also to represent things as complex as contracts and deals among agents.

By contrast with current models of systems regulated by norms [6, 55, 69, 127] in which no distinction among norms is made, our work emphasises that besides the general norms of the system, at least three kinds of norms are needed, namely norms to legislate, to punish, and to reward other agents. By making this differentiation, agents are able to know when an issued norm is valid, when an entitled agent can apply a punishment, and who is responsible for giving rewards. In addition, order is imposed on agents responsible for the normative behaviour of other agents, because their authority is defined by the norms that entitle them to exert social pressure. Roles for *legislators*, *defenders*, and *promoters* of norms become easily identified as a consequence of the different kinds of norms considered. Thus, in this thesis, the authority of agents is always supported and constrained by norms. Table 4.6 compares some of the current models for normative multi-agent systems. This comparison takes into account both the kind of norms included in the models and the concept of authority. In particular, the following elements are assessed: general norms (*GN*); legislation norms (*LN*); enforcement norms (*EN*); rewards norms (*RN*); communication norms (*CN*); and authorities in the system (*AT*). Ticked columns mean that such an element is included in the model, whereas a dash (–) means it is not.

NMAS model	*GN*	*LN*	*EN*	*RN*	*CN*	*AT*
Balzer & Tuomela	✓	✓	–	–	–	–
Esteva & Sierra	✓	–	–	–	✓	–
Moses & Tennenholtz	✓	–	–	–	–	–
Dignum & Dignum	✓	–	–	–	–	–
López	✓	✓	✓	✓	–	✓

TABLE 4.6: Normative Multi-Agent Systems Models

The *dynamics* that occur in a system due to norms have been analysed, and according to the normative behaviour of agents, states for norm instances have been identified. These states enable agents to observe the normative behaviour of other agents and, consequently, they can be used as the elements on which decisions of agents are based. As far as we know, there is no other work in which these dynamics are identified.

So far, we have presented a normative framework that besides providing the means to computationally represent many normative concepts, can be used to give a better understanding of norms and normative agent behaviour. The framework explains not only the role that norms play in the society but also the elements that constitute a norm and that in turn can be used by agents when decisions concerning norms must be taken. By contrast with other proposals, our normative framework has been built upon the idea of *autonomy* of agents. That is, it is intended to be used by agents that must reason about why norms must be adopted, and why an adopted norm must be complied with.

Chapter 5

Agent Powers

5.1 Introduction

One possibility for agents in a society of many agents is that they can overcome their limited capabilities by using the capabilities of others and, in this way, satisfy goals that might not otherwise be achieved. However, given that agents are autonomous, the benevolent adoption of goals cannot be taken for granted since agents can choose not to adopt them [119]. As a result, mechanisms to *convince* agents to adopt goals must be used. Identifying situations in which influence can be exerted is an ability that agents can exploit each time they need to persuade other agents to satisfy their goals. Clearly, agents that have such *power* are more likely to have their goals adopted by agents on whom the power can be exerted.

According to Ott [135], *power* can be defined as the latent ability to *influence* the actions, thoughts or emotions of others and, consequently, it is the potential to get people to do things the way you want them done. Translating these concepts to an agent context, we can say that the powers of an agent are expressed through its abilities to change the beliefs, the motivations, and the goals of other agents in such a way that its goals can be satisfied. Then, power involves a bilateral relationship between agents *with* power and agents on whom the power can be exerted [78]. In an absolutist view, agents without power will never object to the orders of empowered agents and, consequently, their goals are always adopted (i.e. these agents always allow themselves to be influenced.) However, autonomous agents must understand *why* powers should be accepted. The key argument of this chapter is that powers must be accepted if by exerting them the goals of agents without power can be affected. So, agents without power decide when allowing themselves to be influenced is beneficial for their own goals, which is more in accordance with our notion of autonomy.

In agent research, two different approaches to agent powers have been considered, powers that emerge from dependence relationships [2, 26, 31], and powers given to authorities of multi-agent systems organised as social institutions [13, 51, 69, 70, 77, 102, 123]. The first approach corresponds to Castelfranchi's Social Power Theory which states that agents acquire power due to their capabilities to achieve goals that other agents do not have. Consequently, agents without power might agree to satisfy the goals of empowered agents in the hope that these agents help them to satisfy their own goals in the future. The central aspect of Social Power Theory (SPT) is that autonomous agents must have reasons to adopt the goals of others, and these reasons are found in the *dependence* and *power* relationships that emerge among agents. Thus, empowered agents influence dependent agents to satisfy goals.

In the second approach, agents acquire power due to the *authority* they have in a social institution. So, authorities can require members of an institutions to achieve particular goals. In contrast with powers that emerge from an agent's abilities, the recognition of the powers of an authority with a consequent requirement to accept its orders, has *not* been considered as an autonomous decision. In current models of multi-agent systems regulated by norms, the authority of agents is assigned beforehand, is *absolute*, and can never be contravened. Therefore, members of a system are forced to obey an authority's orders, i.e. their autonomy is not respected. Clearly, this does not permit the modelling of agents deciding on their own in which society or relationship they want to participate because they are neither able to object to an authority nor able to understand its *limits*. Although agents that always obey authorities are needed in societies in which absolute control is necessary, they are not suitable for modelling open, dynamic and flexible societies in which autonomous agents that satisfy their own goals can still coexist with other agents.

We argue that agents with abilities to recognise situations of power are needed to model relations of cooperation in which the participation of agents is voluntary. These agents must recognise not only when some powers exist, but also in which situations powers can be exerted in order to be able to constrain powers. As a result, the objective of this chapter is to provide the means for autonomous agents to recognise situations of power in which they might be. Moreover, since motivations for entering and remaining in a society are key aspects in understanding why the authorities of a system are recognised, this aspect will be also considered in the chapter.

Unlike other models in which powers are considered *eternal* and *absolute*, in our model, power is always considered as being *relative* to particular situations of agents and, therefore, *dynamic* because powers appearing in one situation might not exist in another. Our power model always emphasises the autonomy of agents and claims that for autonomous agents to recognise the power of other agents, they must understand how such power might affect their goals.

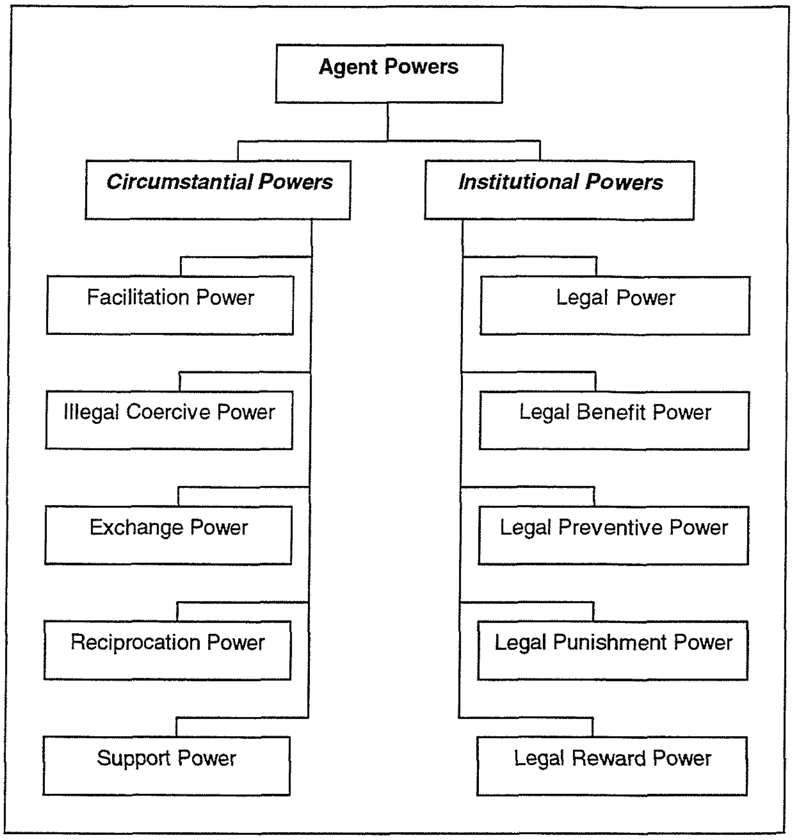


FIGURE 5.1: Powers Taxonomy

Two kinds of powers are identified in this thesis, powers due to an agent's capabilities (*circumstantial powers*), and powers due to the role agents play in the society (*institutional powers*). Figure 5.1 shows a summary of the different situations of powers, identified in this chapter. Circumstantial powers include not only those powers due to agent dependence, but also those powers due to coercive actions that might impede the satisfaction of some goals, and powers that agents *acquire* due to relationships with other agents. Institutional powers are initially accepted by agents when they become part of a society, and continue to be accepted as long as the agents decide to remain in this society. In our view, institutional powers are given by norms that are accepted as legitimate by the members of a society. Thus, the authority of agents is a *legal right* supported by the social structure and, therefore, it must be recognised by all agents that consider themselves as part of the society.

Accordingly with our notion of autonomy, to understand an agent's reasons to enter and stay in a society its goals and motivations must be observed. In particular, we argue that agents enter a society when some of their goals can be satisfied by doing so. However, once inside, agents remain there not only due to the satisfaction of their goals, but also due to the relationships they create with other agents in the society.

The rest of this chapter is organised in four sections as follows. Section 5.2 describes the powers that result from an agent's capabilities (circumstantial powers), while the powers acquired through the norms of a society (institutional powers) are described in Section 5.3. The reasons why autonomous agents enter and stay in a society are explained in Section 5.4. Finally, a summary and conclusions are provided.

5.2 Circumstantial Powers

5.2.1 Introduction

It can be observed that due to their capabilities to satisfy goals, which can contribute to the satisfaction of other agents' goals, agents become empowered because it is their decision to provide or deny the help that other agents might require. That is, agents have the power to facilitate or impede the satisfaction of other agents' goals. Now, if agents are able to satisfy goals that can hinder the goals of other agents, they acquire the power to threaten these agents. In both situations, agents without power are liable to be influenced in order to satisfy their goals. We argue that for power to exist it is not enough to have capabilities to satisfy goals; these goals must have some impact on the goals of other agents.

Some of the powers that appear in this way have been already studied by the *Social Power Theory* [26, 31]. In this section, these and other kinds of powers that are also the result of an agent's abilities are described. In particular, power to facilitate the goals of other agents, power to threaten other agents, power to exchange goals, power of being reciprocated, and power given by supportive agents, are discussed in this section.

5.2.2 Facilitation Power

When the capabilities to satisfy certain goals coincide with the needs of other agents, a basic situation of power emerges: the power to facilitate or impede the goals of other agents. We call it *facilitation power.* In other words, if an agent is able to satisfy a goal for another agent that is unable to satisfy it, we can say that the first agent has power over the second.

In seeking to develop a formal definition of this kind of power, a means to represent the ability to satisfy goals is needed. Sometimes, being able to satisfy a goal is described as being able to execute actions that can lead to the achievement of the goal. In this case the satisfaction of a goal depends on an agent's capabilities. However, being able to satisfy a goal is much complex than this and other aspects must be observed to identify this ability. For instance, some agents are able to satisfy goals even when they are unable to execute all the required actions because it could be the case that they can delegate some actions to others. This is a complex topic and its discussion is beyond the scope of this thesis. Here, the capability of an agent to satisfy a goal in a specific state of the system is formalised via the following predicate.

$$satisfy\_ : \mathbb{P}(NormativeAgent \times Goal \times EnvState)$$

Being able to satisfy goals creates dependence relations between agents with the relevant abilities and those without them. A dependence relationship can be formally defined in terms of an agent's abilities and their absence, as follows. One agent *depends* on another agent if the first agent has a goal that it is unable to satisfy, but the second is able to do so.

$$\begin{array}{l}
depend\_ : \mathbb{P}(NormativeAgent \times NormativeAgent \times Goal \times EnvState) \\
\hline
\forall ag_1, ag_2 : NormativeAgent;\ g : Goal;\ st : EnvState \bullet \\
\quad depend\ (ag_1, ag_2, g, st) \Leftrightarrow (g \in ag_1.goals \land \\
\qquad \neg\ satisfy\ (ag_1, g, st) \land satisfy\ (ag_2, g, st))
\end{array}$$

Now, we can formally define the facilitation power in terms of dependence relationships as follows. An agent has the power to *facilitate* the satisfaction of the goal of other agent, if the second agent depends on the first to satisfy such a goal.

$$\begin{array}{l}
facilitationpower\_ : \mathbb{P}(NormativeAgent \times NormativeAgent \times Goal \\
\qquad \times EnvState) \\
\hline
\forall ag_1, ag_2 : NormativeAgent;\ g : Goal;\ st : EnvState \bullet \\
\quad facilitationpower\ (ag_1, ag_2, g, st) \Leftrightarrow \\
\qquad (g \in ag_2.goals \land depend\ (ag_2, ag_1, g, st))
\end{array}$$

The above relationships are similar to those given by Castelfranchi and colleagues [21, 31, 125]. Detailed definitions of powers and dependence in terms of an agent's plans and capabilities can be found elsewhere [60, 115, 154]. Here, these later definitions are not included because our work is focussed on an agent's goals rather than the actions or the means to achieve those goals.

5.2.3 Illegal Coercive Power

For some agents, their abilities are not used to benefit the goals of other agents, but to impede or hinder them. In these cases, power is expressed by an agent's capabilities to directly threaten the goals of other agents in order to obtain what they want. This power is considered as *illegal* if there is no norm that entitles these agents to coerce the others. This kind of power is generally forbidden, which is why although it is possessed by some agents, it is scarcely used. Formally, we say that an agent has illegal coercive power over another if it is able to satisfy a goal that can hinder one of the goals of the second, the *hinders* predicate is defined in Subsection 3.4.4.

$$\begin{array}{|l}
\mathit{illegalcoercivepower\_} : \mathbb{P}(\mathit{NormativeAgent} \times \mathit{NormativeAgent} \times \mathit{Goal} \\
\qquad\qquad \times \mathit{EnvState}) \\
\hline
\forall\, ag_1, ag_2 : \mathit{NormativeAgent};\ g_2 : \mathit{Goal};\ st : \mathit{EnvState} \bullet \\
\quad \mathit{illegalcoercivepower}\ (ag_1, ag_2, g_2, st) \Leftrightarrow (g_2 \in ag_2.\mathit{goals} \land \\
\qquad (\exists\, g_1 : \mathit{Goal} \bullet (\mathit{satisfy}\ (ag_1, g_1, st) \land \mathit{hinders}(g_1, g_2))))
\end{array}$$

5.2.4 Exchange Power

Castelfranchi et al. [31, 40] state that dependence relationships can give rise to a network of relationships that might be used by agents to influence each other. Among all the possible forms of dependence relationships that Castelfranchi and colleagues identify, *reciprocal dependence* is of particular interest for this thesis because it represents a situation in which two agents need each other (i.e. an agent depends on another to satisfy a goal and *vice versa*), both of them have power, and processes of negotiation might be needed to achieve the goals of each agent. In this particular situation, both agents acquire the so called *exchange power* [47], because both of them have the power to offer something to benefit the goals of the other. Any of the agents can start a negotiation process that finishes with the creation of a social commitment in which each agent receives what it wants. Formally, we say that an agent has exchange power over another agent regarding two specific goals in a particular state if the first agent has the power to facilitate a goal of the second and *vice versa*. This is formalised as follows.

$$\begin{array}{|l}
\mathit{exchangepower\_} : \mathbb{P}(\mathit{NormativeAgent} \times \mathit{Goal} \times \mathit{NormativeAgent} \\
\qquad\qquad \times \mathit{Goal} \times \mathit{EnvState}) \\
\hline
\forall\, ag_1, ag_2 : \mathit{NormativeAgent};\ g_1, g_2 : \mathit{Goal};\ st : \mathit{EnvState} \bullet \\
\quad \mathit{exchangepower}\ (ag_1, g_1, ag_2, g_2, st) \Leftrightarrow \\
\qquad (g_1 \in ag_1.\mathit{goals} \land g_2 \in ag_2.\mathit{goals} \land \\
\qquad \mathit{facilitationpower}\ (ag_1, ag_2, g_2, st) \land \mathit{facilitationpower}\ (ag_2, ag_1, g_1, st))
\end{array}$$

The above relations of power (facilitation, illegal coercive and exchange powers) are defined in terms of both an agent's current goals and the current state of a system. This means that if either the goals of an agent or the state of the system change, these powers might disappear.

5.2.5 Reciprocation Power

Powers also emerge from relations that have occurred in previous interactions. Reciprocation for previous actions has been considered as one of the key aspects underlying society cohesion [82]. Agents that have worked in support of the goals of others generally expect to receive reciprocal benefits, even if not explicitly agreed. This represents an ethical matter in which agents must show their gratitude to others. In particular, agents that satisfied the goals of others without the necessity of being coerced or rewarded acquire *reciprocation power*. To formally define this kind of power, the state of a normative multi-agent system must be considered because it contains a record of all fulfilled norms. This state is represented in the *NMASState* schema in Subsection 4.6.3. We say that an agent has *reciprocation power* in a normative multi-agent system if the following conditions are satisfied.

- Both agents are members of the same system.
- There is a norm instance addressed to the first agent, which has already been fulfilled, whose benefits were enjoyed by the second agent, not including rewards.
- The second agent must have the power to facilitate one of the goals of the first.

These constraints are formalised as follows.

$$
\begin{array}{l}
\mathit{reciprocationpower\_} : \mathbb{P}(\mathit{NormativeAgent} \times \mathit{NormativeAgent} \\
\qquad \times \mathit{NMASState} \times \mathit{EnvState}) \\
\hline
\forall\, ag_1, ag_2 : \mathit{NormativeAgent};\ \mathit{nmas} : \mathit{NMASState};\ \mathit{st} : \mathit{EnvState} \bullet \\
\quad \mathit{reciprocationpower}\ (ag_1, ag_2, \mathit{nmas}, \mathit{st}) \Leftrightarrow \\
\qquad (ag_1.\mathit{self} \in \mathit{nmas.members} \land ag_2.\mathit{self} \in \mathit{nmas.members} \land \\
\qquad (\exists\, \mathit{in} : \mathit{nmas.allinstances} \bullet \\
\qquad\qquad (ag_1.\mathit{self} \in \mathit{in.addressees} \land \mathit{in} \in \mathit{nmas.fulfillednorms} \land \\
\qquad\qquad ag_2.\mathit{self} \in \mathit{in.beneficiaries} \land \mathit{in.rewards} = \varnothing)) \land \\
\qquad (\exists\, g : ag_1.\mathit{goals} \bullet \mathit{facilitationpower}\ (ag_2, ag_1, g, \mathit{st})))
\end{array}
$$

5.2.6 Support Power

One of the things that makes small groups work well is the relations of support or camaraderie that are created among their members. In this case, agents are empowered because they know that they can receive help from any of the agents in a group, and that they are not committed to give something in exchange. We call this group *supportive agents*. The way in which these kinds of groups are created is complex to define and the topic is beyond this research. So, we just assume that each agent has the means to identify a group of supportive agents. The unique restriction for supportive agents is that they must be normative agents because they must be able to comply with all commitments that benefit other agents in the group. Formally, we say that an agent has *support power* over another if both agents belong to the same group of supportive agents, and the second agent has the power to facilitate a goal required by the first.

$$
\begin{array}{l}
\mathit{supportpower\_} : \mathbb{P}(\mathit{NormativeAgent} \times \mathit{NormativeAgent} \times \mathbb{P}\,\mathit{AgentName} \\
\qquad \times \mathit{Goal} \times \mathit{EnvState}) \\
\hline
\forall\, ag_1, ag_2 : \mathit{NormativeAgent};\ \mathit{supportiveags} : \mathbb{P}\,\mathit{AgentName}; \\
\qquad g : \mathit{Goal};\ \mathit{st} : \mathit{EnvState} \bullet \\
\quad \mathit{supportpower}\ (ag_1, ag_2, \mathit{supportiveags}, g, \mathit{st}) \Leftrightarrow \\
\qquad (\{ag_1.\mathit{self}, ag_2.\mathit{self}\} \subset \mathit{supportiveags} \land \\
\qquad \mathit{facilitationpower}\ (ag_2, ag_1, g, \mathit{st}))
\end{array}
$$

This is the kind of power that might exist in a group of friends. Note that the power can exist even if the first agent does not consider the second agent as a friend. It is enough that the second agent believes the first is among its friends to provide it help. This is not critical and we prefer to consider both agents as part of the same group of supportive agents.

Table 5.1 shows a summary of the prevailing conditions for circumstantial powers to emerge.

Type of Power	Conditions for an agent (Ag_1) to become empowered	Conditions for an agent (Ag_2) to be subject of power
Facilitation Power	• Capabilities to satisfy g.	• g is one of its goals.
Illegal Coercive Power	• Capabilities to satisfy g_1. • g_1 hinders g_2.	• g_2 is one of its goals.
Exchange Power	• g_1 is one of its goals. • Capabilities to satisfy g_2.	• g_2 is one of its goals. • Capabilities to satisfy g_1.
Reciprocation Power	• g_1 is one of its goals. • Addressee of a norm (n). • n is fulfilled. • n does not include rewards.	• Capabilities to satisfy g_1. • Beneficiary of the norm n.
Support Power	• g_1 is one of its goals. • Ag_2 is a supportive agent.	• Capabilities to satisfy g_1. • Ag_1 is a supportive agent.

TABLE 5.1: Circumstantial Powers

5.2.7 Discussion

Circumstantial powers emerge during agent interactions and, as any kind of power, they are neither *eternal* nor *absolute*, i.e. powers are constrained. These powers are relativised to a particular situation in which some goals either receive benefits or are hindered. Therefore, what is true in one situation may not be true in another if an agent's interests, and therefore its goals, change. For example, *exchange power* disappears if one of the goals to be interchanged is no longer considered important.

These forms of power can be exemplified by considering four students in an everyday situation as follows. Sanghee is waiting for the postman at home because she is expecting a very important package. She also wants to go for shopping because there is a sale for only two hours in her favourite shop. Cora, Sanghee's neighbor, wants to make a cake for her friends. She needs some ingredients from the supermarket, but cannot go outside because recently she twisted her ankle and cannot walk. Angie is Sanghees friend, and Angie never refuses when Sanghee asks for help. Alejandro is doing his homework on Sanghee's computer.

g_1 : receive a package	g_2: go shopping
g_3: wait for the postman	g_4: make a cake
g_5: get ingredients from the supermarket	g_6: finish homework
g_7: get the computer back	–

TABLE 5.2: Goal Descriptions

By omitting information that is not relevant, the goals of these agents can be listed as shown in Table 5.2. Now, according to the description of these students' current situation, Table 5.3 shows both the goals and the capabilities associated with each agent. The capability of an agent to satisfy a goal g is denoted by the predicate *satisfy*(g).

Agent	Goals	Agent's capabilities
Sanghee	$\{ g_1,g_2 \}$	$\{ \textit{satisfy}(g_5), \textit{satisfy}(g_7) \}$
Cora	$\{ g_4 \}$	$\{ \textit{satisfy}(g_3) \}$
Alejandro	$\{ g_6 \}$	$\{ \textit{satisfy}(g_3) \}$
Angie	$\{ ... \}$	$\{ \textit{satisfy}(g_5), \textit{satisfy}(g_1) \}$

TABLE 5.3: Agent States

By making some assumptions, the relationships that hold among these agent goals are as follows: g_3 benefits g_1, g_5 benefits g_4, and g_7 hinders g_6. Now, according to both the definition of circumstantial powers and the status of agents shown in Table 5.3, some empowered situations among these agents are identified as follows.

- Since g_3 benefits g_1, Cora has *facilitation power* over Sanghee.
- Since g_5 benefits g_4, Sanghee has *facilitation power* over Cora.
- Due to both previous relationships, Cora and Sanghee have *exchange power* over each other and, therefore, have the power to interchange goals with each other. They can make a deal.
- Goal g_7 hinders g_6 and, therefore, Sanghee has *coercive power* over Alejandro, and can oblige Alejandro to wait for the postman.

- Since Angie considers Sanghee her best friend, Sanghee has *supported power* over Angie, and Sanghee can ask her to get the package.
- Once Angie helps Sanghee to satisfy her goal of getting the package, she acquires the *reciprocation power* in the future.

In this example, if Sanghee correctly identifies her powers, she can choose to exert the power which better suits her interests. This decision will determine the way in which she interacts with the others.

5.3 Institutional Powers

5.3.1 Introduction

It is generally accepted that social structures define power relationships derived from the roles agents play in the social system. In such systems there always exist norms that entitle some agents to direct the behaviour of others. Therefore, as long as agents want to *stay* in such a system, they must recognise the power and, therefore, the authority, of certain agents. These kinds of powers are known as *institutional powers*, a term borrowed from [102].

Institutional powers are powers assigned through norms and that are accepted as legitimate by the members of a society (i.e. these powers are supported by the social structure). These powers enable agents to issue new norms, to claim benefits, to prevent agents from violating norms, to claim rewards, and to punish offenders of norms. However, they cannot be exerted at any time, but only when agents over whom the power can be exerted behave in certain ways. For example, to exercise the power of punishing someone, two conditions are needed. On the one hand, there must be an agent empowered by an *enforcement* norm to apply punishments and, on the other, the agent to be punished must have offended such a norm.

Institutional powers only exist in the context of a normative multi agent system which has been defined as a set of normative agents controlled by a common set of norms in Section 4.5. Among the norms of these kinds of systems, the norms to allow the creation of new norms (*legislation norms*), those directed at encouraging compliance with norms by giving rewards (*reward norms*), and norms directed at enforcing compliance with norms by applying punishments (*enforcement norms*) are important to define institutional powers. Based on these norms five different forms of institutional powers are identified: legal power, legal benefit power, legal preventive power, legal punishment power and legal reward power. Details of each one are provided in next subsections.

5.3.2 Legal Power

Legal power is the kind of power that agents entitled to issue new norms have. The validity of a norm could be questioned, and then rejected for all the members of a normative multi-agent system if the norm is issued by agents without this kind of power. For instance, a manager in a factory has legal power. She gives orders to workers under her control, she exerts the power acquired through the role she plays in the factory, and workers accept her orders because they recognise her authority, and therefore her power, in the social structure of the factory. Orders from other people could be rejected. This kind of power is formally defined in the predicate below, which states that an agent has *legal* power over another if both agents are members of the same normative multi-agent system, and the first agent is an addressee of a legislation norm.

$$\mathit{legalpower}\_ : \mathbb{P}(\mathit{NormativeAgent} \times \mathit{NormativeAgent} \times \mathit{NormativeMAS})$$

$$\begin{array}{l} \forall\, ag_1, ag_2 : \mathit{NormativeAgent};\ \mathit{nmas} : \mathit{NormativeMAS} \bullet \\ \quad \mathit{legalpower}\ (ag_1, ag_2, \mathit{nmas}) \Leftrightarrow \\ \qquad (ag_1.\mathit{self} \in \mathit{nmas.members} \land ag_2.\mathit{self} \in \mathit{nmas.members} \land \\ \qquad (\exists\, lg : \mathit{nmas.legislationnorms} \bullet ag_1.\mathit{self} \in lg.\mathit{addressees})) \end{array}$$

5.3.3 Legal Benefit Power

Agents that expect to receive benefits from a norm whose non-compliance is penalised, are also empowered agents. These agents can satisfy their goals by using the responsibilities of other agents, responsibilities that are acquired through the norms of the system. The benefits for these kinds of agents are guaranteed through the social pressure that agents entitled to apply punishments can exert. For instance, a guest in a B&B has the power to request a clean room from the manager, otherwise the English Tourist Board could be informed, and the corresponding penalties might be applied. So, for this power to exist, the following constraints must be fulfilled. The empowered agent must be a beneficiary of a norm of the system, the agent over whom the power is exerted must be an addressee of such a norm, and there must be an enforcement norm that entitles other agents to punish those agents that do not comply with the original. Thus, the social structure supports less favoured agents which would otherwise be unprotected. Those norms without a corresponding enforcement norm do not give power to their beneficiaries because there is no means of exerting social pressure on addressee agents. There, the expected benefits depend on the *willingness* of addressee agents to comply with the norm, which makes it less likely to get those benefits.

Formally, we say that an agent has *legal benefit power* over another agent regarding a norm in a normative multi-agent system if both agents belong to the same system, the first is a beneficiary of the norm, the second is an addressee of the norm, and there is an enforcement norm for it. This is specified as follows.

$$
\begin{array}{|l}
\mathit{legalbenefitpower\_} : \mathbb{P}(\mathit{NormativeAgent} \times \mathit{NormativeAgent} \times \mathit{Norm} \\
\qquad \times \mathit{NormativeMAS}) \\
\hline
\forall\, ag_1, ag_2 : \mathit{NormativeAgent};\ n : \mathit{Norm};\ \mathit{nmas} : \mathit{NormativeMAS} \bullet \\
\quad \mathit{legalbenefitpower}\ (ag_1, ag_2, n, \mathit{nmas}) \Leftrightarrow (n \in \mathit{nmas.generalnorms} \wedge \\
\qquad ag_1.\mathit{self} \in \mathit{nmas.members} \wedge ag_2.\mathit{self} \in \mathit{nmas.members} \wedge \\
\qquad ag_1.\mathit{self} \in \mathit{n.beneficiaries} \wedge ag_2.\mathit{self} \in \mathit{n.addressees} \wedge \\
\qquad (\exists\, \mathit{en} : \mathit{nmas.enforcenorms} \bullet \mathit{enforces}\ (\mathit{en}, n)))
\end{array}
$$

5.3.4 Legal Preventive Power

The power to prevent agents from dismissing norms is reserved for those agents entitled to defend a norm by applying punishment or by giving rewards. These agents may exert pressure over addressees of a norm by reminding them of the potential punishment if they do not comply with the norm, or by reminding them of the rewards they can lose if the norm is dismissed. Thus, this kind of power is acquired either through enforcement norms or through reward norms. Formally, we say that an agent has *legal preventive power* over another agent regarding a norm in a normative multi-agent system if both agents belong to the same system, the second is an addressee of the norm, and the first is an addressee of either an enforcement norm that enforces the norm, or a reward norm that rewards compliance with the norm. This power is represented in the following predicate.

$$
\begin{array}{|l}
\mathit{legalpreventivepower\_} : \mathbb{P}(\mathit{NormativeAgent} \times \mathit{NormativeAgent} \times \mathit{Norm} \\
\qquad \times \mathit{NormativeMAS}) \\
\hline
\forall\, ag_1, ag_2 : \mathit{NormativeAgent};\ n : \mathit{Norm};\ \mathit{nmas} : \mathit{NormativeMAS} \bullet \\
\quad \mathit{legalpreventivepower}\ (ag_1, ag_2, n, \mathit{nmas}) \Leftrightarrow (n \in \mathit{nmas.generalnorms} \wedge \\
\qquad ag_1.\mathit{self} \in \mathit{nmas.members} \wedge ag_2.\mathit{self} \in \mathit{nmas.members} \wedge \\
\qquad ag_2.\mathit{self} \in \mathit{n.addressees} \wedge \\
\qquad ((\exists\, \mathit{en} : \mathit{nmas.enforcenorms} \bullet \\
\qquad\qquad (\mathit{enforces}\ (\mathit{en}, n) \wedge ag_1.\mathit{self} \in \mathit{en.addressees})) \vee \\
\qquad (\exists\, \mathit{rn} : \mathit{nmas.rewardnorms} \bullet \\
\qquad\qquad (\mathit{rewardnorm}\ (\mathit{rn}, n) \wedge ag_1.\mathit{self} \in \mathit{rn.addressees}))))
\end{array}
$$

5.3.5 Legal Punishment Power

An agent entitled to punish gains power as soon as another agent dismisses a norm. That is, an agent is legally allowed to punish another when it fails to comply with a norm. Thus, legal punishment power is limited to cases of norm violation. Entitlement to punish is only acquired through an enforcement norm, so this avoids the situation in which other agents coerce their peers. For instance, in a factory, only managers are entitled (by norms) to fire workers if their production level decreases, but no worker can do so.

Now, since legal punishment power can only be exerted when a norm becomes unfulfilled, to define it we use the state of a normative multi-agent system (*NMASState*) which, besides containing a record of all norms already activated, includes a record of all fulfilled and unfulfilled norms as defined in Subsection 4.6.3. Formally, we say that in a normative multi-agent system, an agent has *legal punishment power* over another agent regarding a norm if both agents belong to the same system, the first is an addressee of an enforcement norm that enforces the norm, the second is the addressee of the norm, and its corresponding instance has been violated. The predicate below expresses these constraints.

$$
\begin{array}{|l}
\mathit{legalpunishmentpower\_} : \mathbb{P}(\mathit{NormativeAgent} \times \mathit{NormativeAgent} \times \mathit{Norm} \\
\qquad \times \mathit{NMASState}) \\
\hline
\forall\, ag_1, ag_2 : \mathit{NormativeAgent};\ n : \mathit{Norm};\ \mathit{nmass} : \mathit{NMASState} \bullet \\
\quad \mathit{legalpunishmentpower}\ (ag_1, ag_2, n, \mathit{nmass}) \Leftrightarrow (n \in \mathit{nmass.generalnorms} \wedge \\
\qquad ag_1.\mathit{self} \in \mathit{nmass.members} \wedge ag_2.\mathit{self} \in \mathit{nmass.members} \wedge \\
\qquad (\exists\, \mathit{en} : \mathit{nmass.enforcenorms} \bullet \\
\qquad\qquad (\mathit{enforces}\ (\mathit{en}, n) \wedge ag_1.\mathit{self} \in \mathit{en.addressees})) \wedge \\
\qquad (\exists\, \mathit{in} : \mathit{nmass.unfulfillednorms} \bullet \\
\qquad\qquad (\mathit{isnorminstance}\ (\mathit{in}, n) \wedge ag_2.\mathit{self} \in \mathit{in.addressees})))
\end{array}
$$

5.3.6 Legal Reward Power

Once an agent complies with a norm that includes rewards, it acquires the power to claim the offered reward. Thus, the agent has the right to be rewarded by the *promoters* of a a norm (i.e. agents responsible for providing rewards.) This responsibility is only acquired through a reward norm. The power is the result of past compliance with a norm and, similarly to legal punishment power, a record of fulfilled norms is needed to define legal reward power. Formally, an agent has *legal reward power* over another agent regarding a norm in a normative multi-agent system if both agents belong to the same system, the first has already fulfilled an instance of the norm, and the norm has a corresponding reward norm for which the second agent is an addressee. This is represented in the predicate below.

$$\begin{array}{l}
\mathit{legalrewardpower\_} : \mathbb{P}(\mathit{NormativeAgent} \times \mathit{NormativeAgent} \times \mathit{Norm} \\
\quad \times \mathit{NMASState}) \\
\hline
\forall \mathit{ag}_1, \mathit{ag}_2 : \mathit{NormativeAgent};\ n : \mathit{Norm};\ \mathit{nmass} : \mathit{NMASState} \bullet \\
\ \ \mathit{legalrewardpower}\ (\mathit{ag}_1, \mathit{ag}_2, n, \mathit{nmass}) \Leftrightarrow (n \in \mathit{nmass.generalnorms} \land \\
\quad \mathit{ag}_1.\mathit{self} \in \mathit{nmass.members} \land \mathit{ag}_2.\mathit{self} \in \mathit{nmass.members} \land \\
\quad (\exists\, \mathit{in} : \mathit{nmass.fulfillednorms} \bullet \\
\qquad (\mathit{isnorminstance}\ (\mathit{in}, n) \land \mathit{ag}_1.\mathit{self} \in \mathit{in.addressees})) \land \\
\quad (\exists\, \mathit{rn} : \mathit{nmass.rewardnorms} \bullet \\
\qquad (\mathit{rewardnorm}\ (\mathit{rn}, n) \land \mathit{ag}_2.\mathit{self} \in \mathit{rn.addressees})))
\end{array}$$

Table 5.4 shows a summary of institutional powers in a normative multi-agent system (*NMAS*) and the conditions for them to be exerted.

5.3.7 Discussion

To exemplify these forms of powers, some hypothetical norms in a university regarding the accommodation provided to students are shown in Table 5.5.

In this example, the normative multi-agent system consists of all the students either living in a hall or starting their first year and looking for a place in the halls, all members of staff dealing with accommodation problems, and all regulations to control students and staff. Now, by observing the description of the norms in Table 5.5, some of their characteristics can be identified as follows. A is a legislation norm. B is a norm directed at the Accommodation Office (denoted in the examples of Table 5.6 by *ACCO*) whose benefits are enjoyed by all first year students. C is an interlocking norm which is activated in the case of B being unfulfilled. D is an enforcement norm activated when C is violated (to punish ACCO). E represents a reward norm for B, and F is an enforcement norm of E. In Table 5.6, the main components of these norms have been roughly extracted. For example, norm B is activated as soon as a first year student submits an application form. To fulfill such a norm, a room must be assigned to the student, a room in a hotel must be found and paid for if this norm is not complied with, but if it is, ACCO will gain money and the reputation of being a reliable office in the University. The addressee of this norm is the Accommodation Office and the direct beneficiaries are the first year students.

Type of Power	**Conditions for an agent (Ag_1) to become empowered**	**Conditions for an agent (Ag_2) to be subject of power**
Legal Power	• Addressee of a legislation norm in the NMAS.	• Member of the NMAS.
Legal Benefit Power	• Beneficiary of norm *n*. • There is a norm (*en*) that enforces norm *n*.	• Addressee of norm *n*.
Legal Preventive Power	• There is a norm (*ern*) that either enforces or rewards norm *n*. • Addressee of norm (*ern*).	• Addressee of norm (*n*).
Legal Punishment Power	• There is a norm (*en*) that enforces norm *n*. • Addressee of the norm (*en*).	• Addressee of norm (*n*). • Norm *n* is violated.
Legal Reward Power	• There is a norm (*rn*) that rewards norm *n*. • Addressee of norm (*n*). • Norm *n* is fulfilled.	• Addressee of the norm (*rn*).

TABLE 5.4: Institutional Powers

By using the definition of institutional powers and the elements of the norms shown in Table 5.6, the following situations of power can be found.

- The Head of the Accommodation Office has *legal power* over students living in university halls.
- All first-year students have *legal benefit* power over the Accommodation Office when they apply for a place in a university hall.
- The Accommodation Office has *legal reward power* when they assign a place for students.

Name	Content
A	All students living in a university hall must follow the regulations issued by the Head of the Accommodation Office.
B	First year students have a guaranteed place in one of the halls of the university if they apply before a term starts.
C	If a place cannot be given to a first year student, the Accommodation Office must find and pay for a room for the student in a nearby hotel until a place in the halls can be given.
D	If the hotel is not paid a fine will be applied by the University.
E	Students located in a hall must pay a monthly rent until the end of their contract.
F	If the accommodation fee remains unpaid by the end of the month, students will be expelled from the university.

TABLE 5.5: Norms in a University Accommodation Office

	Context	Normative goals	Punishments	Rewards	Addressees	Beneficiaries
A	A norm is needed	Issuing new norms	–	–	Head of ACCO	–
B	Application received	Assigning rooms	Finding a hotel	Gaining money and reputation	ACCO	First year students
C	B is violated	Finding a hotel	Losing money and reputation	–	ACCO	First year students
D	C is violated	Imposing fines	–	–	University	ACCO
E	B is fulfilled	Paying fees	Being expelled	–	Students in a hall	ACCO
F	E is violated	Expelling from the university	–	–	University	Students

TABLE 5.6: Components of Norms in the Accommodation Office

- The University has *legal punishment* power over the Accommodation Office when they fail to provide a place to live for a student.
- The University has *legal punishment* power over all students who fail to pay their accommodation fees.
- The University has *legal preventive* power over all students living in a university hall.

Similarly to circumstantial powers, institutional powers are neither *eternal* nor *absolute*. The authorities of a society are recognised as long as agents consider themselves members which, most of the time, is either due to some of their goals being satisfied simply by being there, or due to the relationships agents create with other agents in the society. However, sometimes agents evaluate their society, or compare it with other societies, in order to know which might be more convenient for the satisfaction of their goals. As a result of this evaluation, agents might emigrate to other societies and, consequently, the norms that until now have influenced them, can be abandoned and authorities can lose their legal power.

5.4 Autonomous Membership of Normative Societies

5.4.1 Introduction

In accordance with our notion of autonomy, autonomous agents must express their preferences for being part of a particular relationship, group, organisation or society. Thus, agent motivations are the key to understand why agents join and stay in a society. These motivations also allow us to explain why agents recognise the power and authority of others, and why they adopt and comply with the norms of a society. As long as agents want to stay in a society, they will respect both its authorities and its norms.

Agents join new societies as a means to achieve some of their individual goals. For example, workers join a factory because the money they earn can be used to satisfy their personal goals. As a result, they respect their superiors, adopt the norms of the company and commit themselves to obey those norms. Students join a university in order to satisfy their particular goal of receiving a degree which, in turn, becomes the main motivation to comply with all the university regulations. Software agents that search information in large private databases must agree, for instance, to respect the norms of confidentiality and copyright, before being allowed to access the required information.

However, once agents are in a society the satisfaction of their goals is not the only reason why they stay there. Sometimes, agents acquire certain responsibilities that cannot be dismissed as soon as they achieve their goals. For instance, an agent that joins a credit bureau to get money and, therefore, to satisfy its personal goals, cannot leave the bureau until it fulfills its commitment to repay the money it borrowed. The following subsections are aimed at modelling an agent's decisions to enter and to stay in a society.

5.4.2 Becoming a Member

As mentioned before, autonomous agents join societies because some of their goals can be satisfied by being in those societies [38, 39]. However, since agents might have many goals, and some of them can conflict with the norms of these societies, agents must evaluate the effects on their goals of such membership. *Being* in a society means, on the one hand, that agents have responsibilities acquired through the norms addressed to them and, on the other, that they receive some contributions to their goals from the responsibilities of other agents. Consequently, to decide whether belonging to a society is worthwhile, an agent must assess its responsibilities in a society and the contributions to its goals that the society might offer.

To formally define the terms mentioned above, first, a function to obtain all society norms addressed to a particular agent, is defined as follows.

$$\begin{array}{|l}
normstocomply : (AgentName \times NormativeMAS) \rightarrow \mathbb{P}\, Norm \\
\hline
\forall\, ag : AgentName;\ nmas : NormativeMAS;\ nms : \mathbb{P}\, Norm \bullet \\
\quad normstocomply\ (ag, nmas) = nms \Leftrightarrow (\forall\, n : nms \bullet \\
\qquad (ag \in n.addressees \land n \in nmas.generalnorms))
\end{array}$$

Now, a set of useful functions to extract normative goals, punishments and rewards from a set of norms are defined respectively as follows.

$$\begin{array}{|l}
normgoals, punishgoals, rewardgoals : \mathbb{P}\, Norm \rightarrow \mathbb{P}\, Goal \\
\hline
\forall\, ns : \mathbb{P}\, Norm \bullet \\
\qquad normgoals\ ns = \bigcup\{n : ns \bullet n.normativegoals\} \land \\
\qquad punishgoals\ ns = \bigcup\{n : ns \bullet n.punishments\} \land \\
\qquad rewardgoals\ ns = \bigcup\{n : ns \bullet n.rewards\}
\end{array}$$

Then, the *responsibilities* of an agent in a society are defined as all the goals that must be satisfied by the agent as long as it is considered a member. Formally, the responsibilities of an agent are the normative goals of all the norms for which the agent is an addressee.

$$\begin{array}{|l}
agentresponsibilities : (AgentName \times NormativeMAS) \rightarrow \mathbb{P}\, Goal \\
\hline
\forall\, ag : AgentName;\ nmas : NormativeMAS;\ ngs : \mathbb{P}\, Goal \bullet \\
\quad agentresponsibilities(ag, nmas) = normgoals\ (normstocomply\ (ag, nmas))
\end{array}$$

Agents can obtain contributions to their goals in two ways. They can receive direct contributions as beneficiaries of norms addressed to other agents, or they can receive contributions from the rewards of the norms they ought to fulfill. To formalise this, first, a function to find all the norms of a society from which some benefits are obtained by an agent is defined as follows.

$$\begin{array}{l} \mathit{normsthatbenefit} : (\mathit{AgentName} \times \mathit{NormativeMAS}) \rightarrow \mathbb{P}\,\mathit{Norm} \\ \hline \forall\, ag : \mathit{AgentName};\ nmas : \mathit{NormativeMAS};\ nms : \mathbb{P}\,\mathit{Norm} \bullet \\ \quad \mathit{normsthatbenefit}\,(ag, nmas) = nms \Leftrightarrow (\forall\, n : nms \bullet \\ \qquad (ag \in n.\mathit{beneficiaries} \wedge n \in nmas.\mathit{generalnorms})) \end{array}$$

Formally, the *contributions* that an agent can obtain for being a society member are the normative goals of those norms that benefit the agent, together with the rewards that can be obtained from the norms it has to comply with. Its formalisation is given below.

$$\begin{array}{l} \mathit{societycontributions} : (\mathit{AgentName} \times \mathit{NormativeMAS}) \rightarrow \mathbb{P}\,\mathit{Goal} \\ \hline \forall\, ag : \mathit{AgentName};\ nmas : \mathit{NormativeMAS};\ ngs : \mathbb{P}\,\mathit{Goal} \bullet \\ \quad \mathit{societycontributions}(ag, nmas) = \\ \qquad \mathit{normgoals}\,(\mathit{normsthatbenefit}\,(ag, nmas)) \cup \\ \qquad \mathit{rewardgoals}\,(\mathit{normstocomply}\,(ag, nmas)) \end{array}$$

Now, to evaluate how their responsibilities and the society contributions may affect their goals, agents must consider two things.

- The contributions must provide some benefits for their important goals because it does not make any sense to enter a society from which benefits cannot be taken.
- The new responsibilities must not hinder goals which are more important than the goals that benefit from contributions. Otherwise, agents can lose goals which are more important than those that can be satisfied by being in a society.

To formalise these conditions, first, we define a function which, given two sets of goals, provides the goals of the first that can be hindered by any goal of the second. Here, *hinders* is a predicate that is true when one of the goals hinders the other as explained in Subsection 3.4.4.

$$\begin{array}{l} \mathit{hindered} : (\mathbb{P}\,\mathit{Goal} \times \mathbb{P}\,\mathit{Goal}) \rightarrow \mathbb{P}\,\mathit{Goal} \\ \hline \forall\, gs_1, gs_2, gs_3 : \mathbb{P}\,\mathit{Goal} \bullet \\ \quad \mathit{hindered}\,(gs_1, gs_2) = gs_3 \Leftrightarrow (\forall\, g_1 : gs_3 \bullet \\ \qquad ((g_1 \in gs_1) \wedge (\exists\, g_2 : gs_2 \bullet \mathit{hinders}\,(g_2, g_1)))) \end{array}$$

Now, a function which, given two sets of goals, provides the goals of the first that can benefit from any goal of the second is defined below. As explained in Subsection 3.4.4, the predicate *benefits* is true when a goal benefits another.

$$
\begin{array}{|l}
\mathit{benefited} : (\mathbb{P}\,\mathit{Goal} \times \mathbb{P}\,\mathit{Goal}) \rightarrow \mathbb{P}\,\mathit{Goal} \\
\hline
\forall\, gs_1, gs_2, gs_3 : \mathbb{P}\,\mathit{Goal} \bullet \\
\quad \mathit{benefited}\,(gs_1, gs_2) = gs_3 \Leftrightarrow (\forall\, g_1 : gs_3 \bullet \\
\qquad ((g_1 \in gs_1) \wedge (\exists\, g_2 : gs_2 \bullet \mathit{benefits}\,(g_2, g_1))))
\end{array}
$$

As defined in Section 3.5.2, the *importance* of goals is determined by the intensity of the motivations that are associated with these goals. The higher the intensity of its motivations, the more important the goal. Thus, agent motivations are key aspects for deciding when joining a society is worthy. A normative agent that has chosen the societies in which it wants to stay is formalised in the schema below. There, *societies* is a variable that represents the set of all societies for which the agent is a current member.

$$
\begin{array}{|l}
\hline
\mathit{SocietiesAgent} \\
\hline
\mathit{NormativeAgent} \\
\mathit{societies} : \mathbb{P}\,\mathit{NormativeMAS} \\
\hline
\forall\, s : \mathit{societies} \bullet \mathit{self} \in s.\mathit{members} \\
\hline
\end{array}
$$

Now, every time an agent decides to join a new society, the evaluation of both its responsibilities and the contributions it can receive, must be carried out as follows. Agents evaluate the goals that can be *hindered* by their responsibilities and the goals that can *benefit* from the contributions they receive from the society. Then, the goals that benefit from society contributions must be more important than the goals hindered by an agent responsibilities. We call this constraint the *social satisfaction* condition. If this condition is fulfilled, the agent enters the society, and it must adopt the corresponding society norms as an act of willingness to comply with these norms. However, this does not mean that these norms will be complied with, since the motivations of an agent might change at the moment that these norms must be fulfilled. The process that represents an agent's decision to enter a new society is represented in the *BecomingMember* schema. There, *newsociety?* represents the society that the agent is considering joining. The first predicate states that the agent has not yet entered the new society. The second predicate is the social satisfaction condition evaluated in the *newsociety*. The third predicate represents the agent accepting the society by including it in the set of societies to which the agent belongs. Finally, the last predicate represents the agent adopting the norms of the accepted society.

$$
\begin{array}{|l}
\textit{BecomingMember} \\
\hline
\Delta \textit{SocietiesAgent} \\
\textit{newsociety}? : \textit{NormativeMAS} \\
\hline
\textit{newsociety}? \notin \textit{societies} \\
\textbf{let } \textit{scgs} == \textit{societycontributions}\ (\textit{self}, \textit{newsociety}?) \bullet \\
\textbf{let } \textit{args} == \textit{agentresponsibilities}\ (\textit{self}, \textit{newsociety}?) \bullet \\
\qquad (\textit{importance}\ (\textit{gms}, \textit{benefited}\ (\textit{goals}, \textit{scgs})) \geq \\
\qquad\qquad \textit{importance}\ (\textit{gms}, \textit{hindered}\ (\textit{goals}, \textit{args}))) \\
\textit{societies}' = \textit{societies} \cup \{\textit{newsociety}?\} \\
\textit{norms}' = \textit{norms} \cup \textit{normstocomply}\ (\textit{self}, \textit{newsociety}?) \\
\hline
\end{array}
$$

5.4.3 Society Normative Agent

Before describing how the decision to remain in a society is made, agents able to observe the society in which they participate, and to keep records of past compliance with norms, must be modelled. To do so, we introduce the *SocietyNormativeAgent* schema. This schema includes all the components of the *NormativeAgent* schema, a variable (*society*) to represent the model the agent has about its society, and a variable (*societystate*) which holds records of past compliance with norms in the society (as defined in Subsection 4.6.3). The first predicate in the schema states that the agent considers itself as part of the society, whereas the second states that the agent has adopted some of the norms of this society. The third predicate states that the *societystate* variable must correspond to the state of the agent's society. Finally, the last predicate makes it clear that, when needed, the state of the society, represented by its variable *environment*, must be interpreted according to the agent's beliefs. This raises the possibility that other agents have a different view of their environment and, therefore, inconsistencies among agents can arise. (How to deal with these inconsistencies is beyond the scope of this thesis, and will not be discussed further.)

$$\begin{array}{|l}
\multicolumn{1}{l}{\mathit{SocietyNormativeAgent}} \\
\hline
\mathit{NormativeAgent} \\
\mathit{society} : \mathit{NormativeMAS} \\
\mathit{societystate} : \mathit{NMASState} \\
\hline
\mathit{self} \in \mathit{society.members} \\
\mathit{norms} \cap \mathit{society.generalnorms} \neq \emptyset \\
\mathit{society.nmasname} = \mathit{societystate.nmasname} \\
\mathit{society.environment} = \mathit{beliefs} \\
\hline
\end{array}$$

5.4.4 Staying in a Society

Once agents are in a society, the satisfaction of their important goals is not the only reason why they remain there. Humans, for example, do not emigrate to other societies because of any of the following reasons [35].

- There are no other societies to go, or at least they do not know them.
- They are not willing to face the risk of changing because they are unable to know or predict how their goals could be affected by being there.
- They are under coercion or threats of other humans.
- They have moral commitments to fulfill.
- Their goals are being satisfied in the society.
- They are involved in a network of relations that make them feel strong.
- They need a familiar culture, and they have strong social ties.

Regarding agents, we divide the reasons for staying in a society into two groups: the first includes reasons regarding an agent's goals, and the second includes reasons regarding an agent's relationships. The first group corresponds to those reasons that cause the agent to enter the society. That is, as soon as the important goals of agents continue being satisfied, and their responsibilities do not hinder these important goals, agents will stay there.

The first group of reasons is formally represented in the *StayingbyGoals* schema. There, the *SocietyNormativeAgent* schema is included to represent a normative agent that has entered a society from which some norms have been adopted, and it has models of its society (*society*) and the state of this society (*societystate*). The predicate states that the society being considered satisfies the *social satisfaction* condition as explained earlier.

$$\begin{array}{|l}
\multicolumn{1}{l}{\_\_ StayingbyGoals \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_} \\
SocietyNormativeAgent \\
\hline
\textbf{let } scgs == societycontributions\ (self, society) \bullet \\
\textbf{let } args == agentresponsibilities\ (self, society) \bullet \\
\quad (importance\ (gms, benefited\ (goals, scgs)) \geq \\
\qquad importance\ (gms, hindered\ (goals, args))) \\
\hline
\end{array}$$

In the second group of reasons, an agent assesses its relationships with other agents. Thus, an agent might decide to stay in a society in any of the following cases.

- The agent is being coerced by a member of the society to remain there.
- The agent is part of a group of supportive agents and one of them, which is also a member of the society, needs its help.
- The agent feels obliged to reciprocate to some agents in the society.
- The agent has already decided to comply with norms but their fulfillment has not yet occurred.

In the first three cases, the agent recognises the circumstantial powers of some members of the society. In the last case, the agent is being consistent with the normative decisions it has made. We formalise each reason separately as follows.

Despite the *social satisfaction* condition not being fulfilled, an agent stays in a society if there is someone in the society with the capability of impeding one of the agent's goals that is more important than the goals hindered by the agent's responsibilities. If the agent is not a member of the society, this implies that this important goal cannot be satisfied. This means that although an agent's responsibilities are more than the social contributions the agent can receive, there is a more important goal that the agent can lose if it decides to abandon the society. The formal representation of this is given in the *StayingbyCoercion* schema. There, the *SocietyNormativeAgent* schema is included to represent a normative agent that recognises itself as a member of a society for which a model is held (*society*). The predicate in the schema is composed of several predicates stating the following. First, the social satisfaction condition is not fulfilled in the society being considered. Second, the agent believes that there is a goal regarding which other agent, which is a member of the society, has illegal coercive power over it. Third, such a goal is more important than the goals hindered by the agent's responsibilities. Finally, the fact that the agent is not a member of the society implies that the goal is not among its goals.

$$
\begin{array}{|l}
\multicolumn{1}{l}{\mathit{StayingbyCoercion}} \\
\hline
\mathit{SocietyNormativeAgent} \\
\hline
\textbf{let } \mathit{scgs} == \mathit{societycontributions}\ (\mathit{self}, \mathit{society}) \bullet \\
\textbf{let } \mathit{args} == \mathit{agentresponsibilities}\ (\mathit{self}, \mathit{society}) \bullet \\
\textbf{let } \mathit{bgoals} == \mathit{benefited}\ (\mathit{goals}, \mathit{scgs}) \bullet \\
\textbf{let } \mathit{hgoals} == \mathit{hindered}\ (\mathit{goals}, \mathit{args}) \bullet \\
\quad (\mathit{importance}\ (\mathit{gms}, \mathit{bgoals}) < \mathit{importance}\ (\mathit{gms}, \mathit{hgoals}) \land \\
\quad (\exists\, g : \mathit{goals} \bullet \\
\qquad (\exists\, \mathit{ag} : \mathit{society.members} \bullet \mathit{illegalcoercivepower}\ (\mathit{normativeAg}\ \mathit{ag}, \\
\qquad\qquad \mathit{normativeAg}\ \mathit{self}, g, \mathit{beliefs}) \land \\
\qquad \mathit{importance}\ (\mathit{gms}, \{g\}) \geq \mathit{importance}\ (\mathit{gms}, \mathit{hgoals}) \land \\
\qquad \mathit{self} \notin \mathit{society.members} \Rightarrow g \notin \mathit{goals}))) \\
\hline
\end{array}
$$

An agent stays in a society if it is part of a group of supportive agents, and one of them, which is also a member of the same society, needs its help. We use the *StayingbyFriends* schema to formalise this case. There, the components of the *SocietyNormativeAgent* schema are included as in previous schemas. In addition, a variable (*friends*) to represent the group of supportive agents is included. The predicate in the schema states that there is a member of the society, which is a also a member of the agent's supportive group, and that has a goal that can be satisfied by the agent (i.e. its friend has *supported power*).

$$
\begin{array}{|l}
\multicolumn{1}{l}{\mathit{StayingbyFriends}} \\
\hline
\mathit{SocietyNormativeAgent} \\
\mathit{friends} : \mathbb{P}\, \mathit{AgentName} \\
\hline
\exists\, \mathit{ag} : \mathit{society.members} \bullet (\mathit{ag} \in \mathit{friends} \land \\
\quad (\exists\, g : (\mathit{normativeAg}\ \mathit{ag}).\mathit{goals} \bullet \\
\qquad \mathit{supportpower}\ (\mathit{normativeAg}\ \mathit{ag}, \mathit{normativeAg}\ \mathit{self}, \mathit{friends}, \\
\qquad\qquad\qquad\qquad\qquad\qquad g, \mathit{beliefs}))) \\
\hline
\end{array}
$$

An agent remain in a society when it has to reciprocate another agent for actions from which the former agent was benefited . This case is formalised in the *StayingtoReciprocate* schema where the *SocietyNormativeAgent* schema is also included. The predicate states that the agent believes that there is a member of the society that is expecting to be reciprocated. That is, one of the members of the society has *reciprocationpower* over the agent.

$$
\begin{array}{|l}
\multicolumn{1}{l}{\mathit{StayingtoReciprocate}} \\
\hline
\mathit{SocietyNormativeAgent} \\
\hline
\exists\, ag : society.members \bullet \\
\quad reciprocationpower\ (normativeAg\ ag, normativeAg\ self, \\
\hfill societystate, beliefs) \\
\hline
\end{array}
$$

Finally, an agent stays in a society when it has already decided to comply with norms that have not yet been fulfilled. Here, the agent shows its respect for the commitments it has with other agents. This case is formalised in the *StayingtoComply* schema where, besides the *SocietyNormativeAgent* schema, the *intendednorms* variable is included to represent the norms the agent has decided to comply with. The predicate states that there is a norm of the society the agent has decided to comply with (i.e. it is an intended norm) that has not yet been fulfilled.

$$
\begin{array}{|l}
\multicolumn{1}{l}{\mathit{StayingtoComply}} \\
\hline
\mathit{SocietyNormativeAgent} \\
intendednorms : \mathbb{P}\, NormInstance \\
\hline
\exists\, n : intendednorms \bullet (n \in societystate.allinstances \land \\
\quad n \notin societystate.fulfillednorms) \\
\hline
\end{array}
$$

All these cases can be combined through logical disjunctions to represent an agent's decisions to stay in a society due to the relationships (or ties) it has with other agents in the society as follows.

$$
\begin{array}{l}
StayingbyTies \mathrel{\widehat=} StayingbyCoercion \lor StayingbyFriends \lor \\
\qquad\qquad StayingtoReciprocate \lor StayingtoComply
\end{array}
$$

5.5 Conclusions

On the one hand, agents must be able to recognise their powers in order to use them to cause agents, on whom the power can be exerted, to satisfy their goals. On the other hand, agents must be able to recognise the power of other agents in order to know when the goals of those agents can be adopted, and to be able to work in systems in which the authority of agents exists.

Two kinds of power are identified in this chapter: powers due to an agent's capabilities, and powers acquired through the roles agents play in a society. These are called *circumstantial* powers and *institutional* powers, respectively. Contrary to much work in which powers of agents are taken as eternal and absolute, all situations of power have been relativised to an agent's goals and to the society in which agents exist. This allows agents to decide, at run-time and according to their current situation, both if someone's authority is valid and if a particular kind of power can be exerted. This also enables agents to constrain the power of agents and to avoid abusive situations from emerging.

Circumstantial powers are those that result from an agent's abilities and the circumstances of others that cause the agent to be in an empowered situation. Although our analysis builds on important work on power, dependence and norms, it goes beyond *power due to dependence* [21, 26, 31] by including other powers not considered before. In particular, five types of circumstantial powers are identified in this chapter: power to facilitate the goals of other agents, power to threaten other agents, power to exchange goals, power of being reciprocated, and power given by relations of support among agents.

As long as an agent wants to stay in a society, the powers and authority of certain agents must be recognised. We define institutional powers as the powers supported by the norms in a society. In this chapter, five empowered situations due to norms have been identified: power to legislate and to issue new norms, power to claim the benefits that some norms provide to agents, power to prevent agents deviating from norms, power to punish offenders, and power to reward compliant agents. Some of these relations of power due to norms have been mentioned in other work [146, 157], but the full set identified here has not been considered as a coherent whole.

As mentioned, to understand why agents recognise the authority and, therefore, the power of certain agents in a society, the reasons for agents to enter and to stay in a society are explained. We argue that autonomous agents enter a society as a means of satisfying their important goals. However, since their other goals might be in conflict with the norms of a society, agents must evaluate the contributions they can get from being members against the responsibilities they have in the society. In addition, once inside a society, agents stay there not only to satisfy their goals but also due to the relationships that they create with other agents in the society. In particular, we argue that an agent stays in a society because of any of the following reasons: its goals are being satisfied, someone is threatening the agent, some of its friends need help, the agent wants to reciprocate others from whom help was received, or the agent has committed itself to comply with some norms. The problems of entering and leaving a society have been previously treated from the perspective of the society, organisation or group in consideration. That is, there is always someone that selects agents as members of a society on the basis of their capabilities, among other things [7, 48, 51, 55, 144]. Our perspective is complementary to all these approaches because in this case, it is the agent who evaluates if membership is worthy for its own interests.

López	**Ross**	**Singh**
Power to legislate	Competence	–
Under power to legislate	Subjection	–
Legal benefit power	Claim	Claim
Under legal benefit power	Obligation	Commitment
Legal preventive power	Obligation	Power
Legal punishment power	–	–
Legal reward power	–	–

TABLE 5.7: Institutional Power Comparison

Table 5.7 compares the situations of institutional power identified in this chapter (denoted by *López* in the table), against similar normative relations presented by Ross [146] and Singh [157]. Note that the compared models are taken from different, yet compatible, areas of research. Whereas Ross's work is focussed on the formalisations of legal actions, Singh's research is interested in the interactions that might occur in a multi-agent system due to norms. By contrast, our work is focussed on autonomous agents and what they need to take effective decisions regarding norms. From the table, we can observe that our model, besides including the others, allows the identification of powers that result from the normative behaviour of other agents. This is possible thanks to the use of the previously defined normative framework which considers the dynamics that result from the normative behaviour of agents in a system.

Chapter 6

The Impact of Powers on Agent Behaviour

6.1 Introduction

Undoubtedly, one of the cornerstones of agent research is *cooperation* [64]. We understand cooperation as the process by which one or more agents agree to satisfy the goals of other agents [118]. Goals do not simply emerge, but are either generated or owned by agents [119, 178], which might require the cooperation of other agents to satisfy them. However, before requiring the adoption of a goal, an agent must take into account that autonomous agents can decide not to cooperate, and that time can be wasted due to fruitless interactions with agents that ultimately reject such requests. This can be avoided if agents are provided with the means to identify in advance if the adoption of a goal is likely to be accepted. Thus, the number of necessary interactions to reach an agreement to cooperate can be reduced [124, 125]. Moreover, if agents are provided with the means to understand the reasons why other agents should adopt their goals, they may already be prepared to argue in case of rejection [10, 106, 138], and a better selection of plans can be made if agents know in advance which agents they can rely on [84].

Because agents are autonomous, they must be *convinced* to adopt external goals. This can be done via a process of negotiation [10, 72, 128] in which agents take advantage of their situations of *power* to influence other agents to satisfy their goals [112]. However, powers are constrained, and agents must identify in which cases, and over which agents, these powers can be exerted. Conversely, if agents *without* power recognise their vulnerability, they might be influenced to adopt the goals of an empowered agent, so that both agents might participate in a relation of cooperation [62]. Moreover, if the cooperation of other agents is needed to execute a plan, it is desirable to select the plan whose subgoals have lower probability of being rejected by others.

Using agent powers to influence other agents to satisfy goals is a principle of Social Power Theory [21, 27, 31, 36] but, although SPT can be directly applied to situations in which agents know each other's goals and can identify their dependence relationships, it fails to explain those cases of cooperation in which interdependence among agents is not obvious. For example, people on buses give up their seats to the elderly without expecting to receive something in exchange, and nurses in a hospital cooperate with doctors not to interchange goals but as part of their responsibilities in the organisation. By contrast, our view of powers includes not only those due to an agent's abilities to satisfy goals that might either contribute to, or impede, the goals of other agents (*circumstantial* powers), but also those due to the roles agents play in a society (*institutional* powers) (both kinds of power are discussed in Chapter 5.) We argue that if power can be exerted, agents *without power* might accept the adoption of the goal of an agent *with power*. In this way, autonomous agents are *influenced* by other agents to create commitments among them.

Agent powers have a direct impact on three processes of decision-making: delegation of goals, adoption of goals, and selection of plans.

- During goal delegation, powers are used to identify when a goal can successfully be delegated. This occurs when the powers of the delegating agents can be exerted to influence an agent to adopt the goal.
- In goal adoption, powers are used for agents to understand why a goal must be adopted. Here, agents recognise the powers of the delegating agent and how these powers can affect their goals.
- Powers can also be used during the selection of a plan because they are the means to foresee the success of a plan by predicting the success of the delegation of those subgoals, in the plan, that cannot be satisfied by the agent.

So, analysing when powers can be exerted to delegate goals, when the power of other agents must be recognised to adopt their goals, and how to select plans according to the possibilities for delegating subgoals by using agent powers, are the objectives of this chapter. Additionally, since the adoption of external goals represents commitments towards other agents, the non-satisfaction of these goals can be penalised, and their complete satisfaction can be rewarded, we argue that every agreement to satisfy external goals must be *made formal* through the adoption of a corresponding norm. So, the problem of autonomous norm adoption is also faced in this chapter.

To satisfy the objectives of this chapter, in Section 6.2 the conditions that must prevail for agents to believe that one of their goals can be successfully delegated are provided. In the same section, a classification for agents that can be influenced to adopt an external goal is given. Section 6.3 presents a classification of plans based on the possibility of delegating subgoals. Section 6.4 describes, in terms of agent powers, the conditions for agents to believe that external goals must be adopted. In Section 6.5 a model for norm adoption as an autonomous decision is presented. Finally, our conclusions are provided.

6.2 Goal Delegation

6.2.1 Introduction

There are some situations in which agents prefer their goals to be satisfied by others and, according to Castelfranchi and Falcone [30], these agents have three options.

- Agents can take advantage of situations in which other agents are already satisfying the required goal. Then, agents just have to wait until the required goal becomes satisfied. For instance, an agent that is leaving a room and has the goal of opening the door can wait until another agent opens the door to go through.
- Sometimes, it is possible to change the beliefs and motivations of agents in order to provoke the desired goal being generated as one of their goals. That is, by changing an agent's motivations, new needs and, therefore, new goals are generated [117, 118]. For example, a car advertisement affects the motivations of people to make them buy cars of a certain brand.
- The last option is to reach agreements by mutual consent between agents that *delegate* goals and agents willing to *adopt* these goals.

In the first case, agents take advantage of their current situation, and in the second, they work to influence other agents by changing their motivations. In both cases, there are neither direct interactions nor commitments between agents. Consequently, those agents capable of achieving the desired goal are free to intend or drop it. By contrast, the last option involves the decisions of two agents to voluntarily create *commitments* of cooperation. It is the last case that is the concern of this section where the problem of goal delegation is faced. The goal adoption problem is faced later on.

To delegate a goal, an agent must consider the possibilities of influencing the agent chosen to delegate to. This must be done by identifying the kind of power that can be exerted over the agent [112]. In the remainder of this section, we provide the means for agents to identify situations in which, by exerting their circumstantial or institutional powers, others can be influenced to adopt a delegated goal.

6.2.2 Exerting Circumstantial Powers

6.2.2.1 Threatening Agents

Some agents prefer to use coercive methods to influence others to satisfy their goals. To do so, these agents must have the capabilities to hinder the goals of the agents to be threatened, i.e. they must have *illegal coercive* power. However, for the threat to be effective, the agent without power must consider the hindered goal as more important than other goals that could be hindered by satisfying the external agent's goals, otherwise the threat might be ignored. Empowered agents must take into account the motivations of the threatened agents in order to succeed in the delegation of their goals. Formally, an agent that has *illegal coercive power* over another can delegate any of its goals if this illegal coercive power can be exerted to hinder a goal whose importance is higher than the importance of the goals hindered by the delegated goal. This is formally expressed in the relationship below. Here ag_1 represents the agent requiring a goal (g) to be delegated, and ag_2 represents the agent chosen to delegate to.

$$
\begin{array}{l}
\mathit{threatendelegation\_} : \mathbb{P}(\mathit{NormativeAgent} \times \mathit{NormativeAgent} \times \mathit{Goal}) \\
\hline
\forall\, ag_1, ag_2 : \mathit{NormativeAgent};\ g : \mathit{Goal} \bullet \\
\quad \mathit{threatendelegation}\ (ag_1,\ ag_2,\ g) \Leftrightarrow (\exists\, g_1 : ag_2.\mathit{goals} \bullet \\
\qquad (\mathit{illegalcoercivepower}\ (ag_1,\ ag_2,\ g_1,\ ag_1.\mathit{beliefs}) \land \\
\qquad (\mathit{importance}\ (ag_2.\mathit{gms},\ \{g_1\}) > \\
\qquad\quad \mathit{importance}\ (ag_2.\mathit{gms},\ \mathit{hindered}\ (ag_2.\mathit{goals},\ \{g\})))))
\end{array}
$$

6.2.2.2 Exchanging Goals

Delegation of goals can also be achieved through exchange of goals when two agents recognise that they can help each other, and each agrees to satisfy the goal of the other. One of the main difficulties in reaching an agreement in relation of goal exchange relates to the worth of goals [47]. Goals can be measured, for instance, according to their importance or the cost that the satisfaction of a goal might imply. If all goals are equivalent, the deal is always considered fair for both agents and exchange of goals may be rapidly achieved. However, in the majority of cases not all goals are equivalent. This makes the deal unfair for the agent that has to achieve the most difficult, important, or costly goal. Consequently, before an agreement can be made, a process to evaluate (and maybe to negotiate) the goals to be exchanged and the rewards to be offered, is needed.

For instance, workers in a factory may be motivated by a manager to increase their productivity if a promise to earn extra money for each item produced is made. Clearly, *exchange power* exists, since the manager needs workers for the achievement of the goals that have been assigned to her, and workers need the extra money for the satisfaction of their goals. However, both the quantity of items that workers must produce, and the quantity of money they receive, must be negotiated by both parties. The discussion of negotiation techniques is beyond the scope of this thesis and will not be considered further. However, we can still consider the conditions for an agent to believe it can delegate a goal through exchange.

For an agent to initiate the delegation of one of its goals through exchange, it must recognise its *reciprocal dependence* on the agent chosen to achieve the goal (i.e. the agent must have power to exchange goals). Now, since autonomous agents act according to their motivations, the agent must believe that the goal to be exchanged is more important than the goals that may be hindered by the goal to be adopted (through exchange). Moreover, the agent must believe that this condition also applies to the chosen agent. This is important because it does not make any sense to offer the satisfaction of an unimportant goal when other more important goals could be hindered.

Formally, an agent (ag_1) can delegate any goal to another agent (ag_2) if both agents have *exchange power*, the goal to be delegated (g_1) is more important than the goals hindered by the goal to be adopted goal (g_2), and the agent believes that the chosen agent has similar reasons to exchange one of its goals. The representation of this is given below.

$$
\begin{array}{|l}
\mathit{exchangedelegation\_} : \mathbb{P}(\mathit{NormativeAgent} \times \mathit{NormativeAgent} \times \mathit{Goal}) \\
\hline
\forall\, ag_1, ag_2 : \mathit{NormativeAgent};\ g_1 : \mathit{Goal} \bullet \\
\quad \mathit{exchangedelegation}\ (ag_1,\ ag_2,\ g_1) \Leftrightarrow (\exists\, g_2 : ag_2.\mathit{goals} \bullet \\
\qquad (\mathit{exchangepower}\ (ag_1,\ g_1,\ ag_2,\ g_2,\ ag_1.\mathit{beliefs}) \land \\
\qquad (\mathit{importance}\ (ag_1.\mathit{gms},\ \{g_1\}) > \\
\qquad\qquad \mathit{importance}\ (ag_1.\mathit{gms},\ \mathit{hindered}\ (ag_1.\mathit{goals},\ \{g_2\}))) \land \\
\qquad (\mathit{importance}\ (ag_2.\mathit{gms},\ \{g_2\}) > \\
\qquad\qquad \mathit{importance}\ (ag_2.\mathit{gms},\ \mathit{hindered}\ (ag_2.\mathit{goals},\ \{g_1\})))))
\end{array}
$$

6.2.2.3 Requiring Support

Autonomy does not restrict an agent's benevolence. Consider an agent with a group of special agents (such as a group of close friends) for which help is never denied and whose goals are adopted as an end. Neither rewards nor punishments are expected by those agents adopting their goals. So, agents can take advantage of the support of these groups of agents to delegate their goals. Since explaining how these kinds of groups are formed and maintained is a topic beyond this thesis, we assume that each agent has the means to identify which agents are included in them. Formally, we say that an agent that has *support power* can delegate any goal to any agent in a set of *supportive agents*.

$$\begin{array}{|l}
\mathit{supportivedelegation\_} : \mathbb{P}(\mathit{NormativeAgent} \times \mathit{NormativeAgent} \times \\
\qquad\qquad \mathbb{P}\,\mathit{AgentName} \times \mathit{Goal}) \\
\hline
\forall\, ag_1, ag_2 : \mathit{NormativeAgent};\ \mathit{supportiveags} : \mathbb{P}\,\mathit{AgentName};\ g : \mathit{Goal} \bullet \\
\quad \mathit{supportivedelegation}\ (ag_1,\ ag_2,\ \mathit{supportiveags},\ g) \Leftrightarrow \\
\qquad \mathit{supportpower}\ (ag_1,\ ag_2,\ \mathit{supportiveags},\ g,\ ag_1.\mathit{beliefs})
\end{array}$$

6.2.2.4 Requiring Reciprocation

Those agents that have provided some benefits to others, acquire *reciprocation power* to be used when they need it. This could be seen as the counterpart of benevolent adoption because, given that agents have been helped in the past without conditions, they should provide help in the same way. Formally, an agent that has reciprocation power over another agent can delegate it any of its goals. Notice that, in this case, the state of a society (formalised in the *NMASState* schema in Subsection 4.6.3) is needed because a record of past compliance with norms is required to identify when reciprocation power can be exerted.

$$\begin{array}{|l}
\mathit{reciprocatedelegation\_} : \mathbb{P}(\mathit{NormativeAgent} \times \mathit{NormativeAgent} \\
\qquad\qquad \times \mathit{NMASState} \times \mathit{Goal}) \\
\hline
\forall\, ag_1, ag_2 : \mathit{NormativeAgent};\ \mathit{societystate} : \mathit{NMASState};\ g : \mathit{Goal} \bullet \\
\quad \mathit{reciprocatedelegation}\ (ag_1,\ ag_2,\ \mathit{societystate},\ g) \Leftrightarrow \\
\qquad \mathit{reciprocationpower}\ (ag_1,\ ag_2,\ \mathit{societystate},\ ag_1.\mathit{beliefs})
\end{array}$$

6.2.3 Exerting Institutional Powers

6.2.3.1 Issuing Orders

Agents that have power to issue new orders can easily delegate a goal to those agents in their domain of competence because they are recognised as authorities in the society. Formally, an agent that has *legal power* in a society can delegate goals to any of its members. Such delegation can be formalised as follows.

$$
\begin{array}{|l}
\mathit{legaldelegation\_} : \mathbb{P}(\mathit{NormativeAgent} \times \mathit{NormativeAgent} \times \mathit{Goal} \\
\qquad\qquad \times \mathit{NormativeMAS}) \\
\hline
\forall\, ag_1, ag_2 : \mathit{NormativeAgent};\ g : \mathit{Goal};\ \mathit{society} : \mathit{NormativeMAS} \bullet \\
\quad \mathit{legaldelegation}\ (ag_1, ag_2, g, \mathit{society}) \Leftrightarrow g \in ag_1.\mathit{goals} \land \\
\qquad \mathit{legalpower}(ag_1, ag_2, \mathit{society})
\end{array}
$$

6.2.3.2 Applying Punishments

An agent can delegate a goal to another agent if such a goal must be satisfied as part of the punishments of an unfulfilled norm, and the agent delegating the goal is an authority entitled to apply these punishments, i.e. it has *legal punishment power*. This means that the agent required to adopt the goal is an offender of a norm that must now accept being punished. Formally, an agent (ag_1), which has legal punishment power regarding a norm n, can delegate those goals that correspond to the punishments of the norm to any one of its addressees (ag_2 in this case) that did not comply with it. The state of a society (*NMASState*) is needed in the definition because a record of past compliance with norms is required to identify when legal punishment power can be exerted.

$$
\begin{array}{|l}
\mathit{punishdelegation\_} : \mathbb{P}(\mathit{NormativeAgent} \times \mathit{NormativeAgent} \times \mathit{Goal} \\
\qquad\qquad \times \mathit{NMASState}) \\
\hline
\forall\, ag_1, ag_2 : \mathit{NormativeAgent};\ g : \mathit{Goal};\ \mathit{societystate} : \mathit{NMASState} \bullet \\
\quad \mathit{punishdelegation}\ (ag_1, ag_2, g, \mathit{societystate}) \Leftrightarrow \\
\qquad (\exists\, n : \mathit{societystate.generalnorms} \bullet \\
\qquad\qquad (\mathit{legalpunishmentpower}\ (ag_1, ag_2, n, \mathit{societystate}) \land \\
\qquad\qquad g \in n.\mathit{punishments}))
\end{array}
$$

6.2.3.3 Claiming Rewards

An agent can delegate a goal if the goal must be satisfied by another agent as part of a previously offered reward. That is, by complying with norms, agents become entitled to claim its rewards. Formally, an agent (ag_1), which has *legal reward power* regarding a norm n, can delegate those goals that correspond to the rewards of the norm n to any agent (ag_2) required to provide rewards.

$$
\begin{array}{|l}
\mathit{rewarddelegation\_} : \mathbb{P}(\mathit{NormativeAgent} \times \mathit{NormativeAgent} \times \mathit{Goal} \\
\qquad\qquad \times \mathit{NMASState}) \\
\hline
\forall\, ag_1, ag_2 : \mathit{NormativeAgent};\ g : \mathit{Goal};\ \mathit{societystate} : \mathit{NMASState} \bullet \\
\quad \mathit{rewarddelegation}\ (ag_1, ag_2, g, \mathit{societystate}) \Leftrightarrow \\
\qquad (\exists\, n : \mathit{societystate.generalnorms} \bullet \\
\qquad\qquad (\mathit{legalrewardpower}\ (ag_1, ag_2, n, \mathit{societystate}) \land \\
\qquad\qquad\ g \in n.\mathit{rewards}))
\end{array}
$$

6.2.3.4 Claiming Social Benefits

Beneficiaries of a norm can delegate any goal that coincides with the normative goals of a norm to any of its addressees. Formally, an agent (ag_1) that has *legal benefit power* regarding a norm n, can delegate a goal (g) to an agent (ag_2) over whom this power can be exerted, if the goal corresponds to the normative goals of the norm. This formalisation is given below.

$$
\begin{array}{|l}
\mathit{benefiteddelegation\_} : \mathbb{P}(\mathit{NormativeAgent} \times \mathit{NormativeAgent} \times \mathit{Goal} \\
\qquad\qquad \times \mathit{NormativeMAS}) \\
\hline
\forall\, ag_1, ag_2 : \mathit{NormativeAgent};\ g : \mathit{Goal};\ \mathit{society} : \mathit{NormativeMAS} \bullet \\
\quad \mathit{benefiteddelegation}\ (ag_1, ag_2, g, \mathit{society}) \Leftrightarrow \\
\qquad (\exists\, n : \mathit{society.generalnorms} \bullet \\
\qquad\qquad (\mathit{legalbenefitpower}\ (ag_1, ag_2, n, \mathit{society}) \land \\
\qquad\qquad\ g \in n.\mathit{normativegoals}))
\end{array}
$$

6.2.4 Discussion

All of the above forms of identifying situations in which circumstantial powers can be exerted to delegate a goal require the agent to have a model of the goals, beliefs and motivations of other agents. By contrast, to exert institutional powers agents do not need a model of the others because, through norms, the responsibilities of each agent are very well established and agents expect those responsibilities to be fulfilled. That is, agents use the support of the society where they participate to delegate their goals. It may seem that the delegation of goals can be guaranteed by exercising institutional powers; however, since agents are autonomous, they can decide at any time to ignore those powers and leave the society, organisation or group.

Finally, it is important to say that these forms of identifying when other agents can be influenced are interpreted according to the beliefs of the delegating agent which means that the delegation of a goal cannot succeed if the beliefs of the other agent do not coincide with them.

6.3 Plan Selection Criteria

6.3.1 Introduction

After an agent has decided what to do, it must decide how it must be done. In other words, once a goal has been selected to be achieved, the means to satisfy it must be chosen. In general, these means are represented as plans whose execution might require not only the abilities of the agent, but also the abilities of other agents. Choosing a plan in terms of the capabilities of other autonomous agents is risky because these others may decide not to provide help. Consequently, the possibility for agents to delegate goals must be considered when selecting a plan [63, 124, 125]. That is, agents must identify beforehand if those agents that might be involved in the execution of a plan, can be influenced in some way.

Some basic heuristics to select a plan based on the number of involved actions, their cost, and the required time have already been provided by others [147]. Work has also been done regarding the selection of plans that involve the actions of benevolent agents (e.g. [66, 67]). There are also strategies to select plans based on factors of risk introduced by the degree of trust in agents that have previously agreed to cooperate [84], and strategies based on relations of dependence [50]. In addition, Luck and d'Inverno analyse plans in terms of both the number of agents required to cooperate, and the number of agents that might be affected by the execution of a plan [121]. By contrast, our proposal is to analyse a plan on the basis of the *kinds of influence* that can be used with agents involved in its execution, which depend on the kinds of powers that can be exerted over these agents. This analysis provides the means to classify those plans that require the cooperation of other agents in order to provide alternatives to satisfy a particular goal. Figure 6.1 shows the seven categories of plans identified in this section.

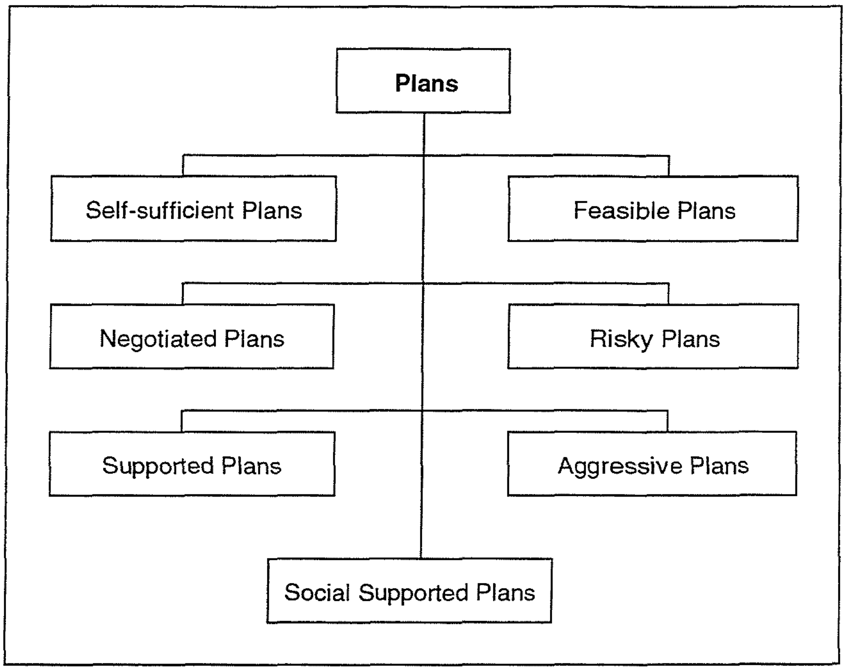


FIGURE 6.1: Plan Categories

6.3.2 Potential Partner Agents

Before categorising plans, a classification of those agents that can be influenced to adopt a particular goal is presented. So, potential partners to delegate goals are those agents that can be coerced to cooperate (*threatened agents*), those with which goal exchange can be agreed (*contractors*), those agents already committed to adopt goals (*captive agents*), and those agents whose responsibilities in a society commit them to adopt the external goals (*institutional partners*). Table 6.1 shows the categories of agents identified in this section, together with the kinds of influence that can be used with them, which depends on the kind of power that can be exerted on these agents. Details of each category are provided below.

6.3.2.1 Circumstantial Partners

Agents that can be influenced to adopt a determined goal, by using circumstantial powers, are divided into three different categories, as follows.

- Those agents that can be threatened to satisfy a goal are gathered in a set of agents called *threatened agents*.
- The second category corresponds to all agents for which a negotiation process might be needed to reach an agreement. That is, agents that can be convinced to adopt a goal in exchange for one of their own goals are called *contractors*.

Type of Agent	**Influence Argument**
Threatened Agents	• Threatening
Contractors	• Exchanging Goals
Captive Agents	• Claiming Support • Requiring Reciprocation
Institutional Partners	• Issuing Orders • Applying Punishments • Claiming Rewards • Claiming Social Benefits

TABLE 6.1: Potential Partner Agents

- Agents that can adopt external goals, either as a favour for other agents or in order to offer reciprocation to agents from whom help was received in the past, constitute the next category of partner agents. We call these agents *captive agents*.

In the *CircumstantialPartners* schema, we introduce an agent able to classify other agents over whom circumstantial powers can be exerted to influence them to adopt external goals. There, the *SocietyNormativeAgent* schema is included to represent a normative agent that has a model of the society in which it participates. In addition, the following variables are introduced: the *acquaintances* variable which represents all agents the agent knows about, and the *supportiveags* variable which represents the agent's group of supportive agents. In the schema, there are also three different functions: *threatenags*, *contractorags*, and *captiveags* which, given a goal, provide the set of agents considered as potential agents to adopt such a goal according to the different criteria that might influence them. The first predicate in the schema states that some of the members of the society are acquaintances of the agent. The second predicate states that the set of supportive agents must all be acquaintances of the agent. The third predicate is the definition of the *threatenags* function which provides the set of agents that can be threatened to adopt a goal. The *contractorags* function is defined in the fourth predicate, and provides the set of agents that can adopt a goal if something is offered in exchange. Finally, the fifth predicate is the definition of *captiveags* function, which provides the set of agents that can adopt a goal either to provide support to the agent or to reciprocate given past actions.

$$
\begin{array}{|l}
\multicolumn{1}{l}{\mathit{CircumstantialPartners}} \\
\mathit{SocietyNormativeAgent} \\
\mathit{acquaintances} : \mathbb{P}\,\mathit{AgentName} \\
\mathit{supportiveags} : \mathbb{P}\,\mathit{AgentName} \\
\mathit{threatenags} : \mathit{Goal} \nrightarrow \mathbb{P}\,\mathit{AgentName} \\
\mathit{contractorags} : \mathit{Goal} \nrightarrow \mathbb{P}\,\mathit{AgentName} \\
\mathit{captiveags} : \mathit{Goal} \nrightarrow \mathbb{P}\,\mathit{AgentName} \\
\hline
\mathit{society.members} \cap \mathit{acquaintances} \neq \varnothing \\
\mathit{supportiveags} \subset \mathit{acquaintances} \\
\forall\, g : \mathit{Goal} \bullet \mathit{threatenags}\ (g) = \{ \mathit{ag} : \mathit{acquaintances} \mid \\
\quad \mathit{threatendelegation}\ (\mathit{normativeAg\ self}, \mathit{normativeAg\ ag}, g) \} \\
\forall\, g : \mathit{Goal} \bullet \mathit{contractorags}\ (g) = \{ \mathit{ag} : \mathit{acquaintances} \mid \\
\quad \mathit{exchangedelegation}\ (\mathit{normativeAg\ self}, \mathit{normativeAg\ ag}, g) \} \\
\forall\, g : \mathit{Goal} \bullet \mathit{captiveags}\ (g) = \{ \mathit{ag} : \mathit{acquaintances} \mid \\
\quad (\mathit{supportivedelegation}\ (\mathit{normativeAg\ self}, \mathit{normativeAg\ ag}, \\
\qquad\qquad\qquad\qquad\qquad\qquad \mathit{supportiveags}, g) \vee \\
\quad \mathit{reciprocatedelegation}\ (\mathit{normativeAg\ self}, \mathit{normativeAg\ ag}, \mathit{societystate}, g)) \} \\
\hline
\end{array}
$$

6.3.2.2 Institutional Partners

Those agents that can be influenced to adopt a delegated goal by invoking their responsibilities in a society regulated by norms are called *institutional partners.* That is, an agent can delegate a goal to any member of a set of agents if it is entitled to issue orders, to punish, or to claim rewards, or if it is the beneficiary of a norm. As previously stated, this way of delegating goals is always supported by the social structure.

In the *InstitutionalPartners* schema, we introduce an agent able to classify other agents over whom institutional powers can be exerted to influence them to adopt external goals. There, the *SocietyNormativeAgent* schema is included to represent a normative agent that has a model of the society in which it participates. A function (*institutionalags*) which, given a goal, provides the set of agents to which the goal can be delegated as an order, as applying a punishment, as receiving rewards or as claiming the benefits of a norm is also included. The definition of the function is specified in the unique predicate.

$$
\begin{array}{|l}
\multicolumn{1}{l}{\underline{\mathit{InstitutionalPartners}}} \\
\mathit{SocietyNormativeAgent} \\
\mathit{institutionalags} : \mathit{Goal} \rightharpoonup \mathbb{P}\, \mathit{AgentName} \\
\hline
\forall g : \mathit{Goal} \bullet \mathit{institutionalags}\ (g) = \{ \mathit{ag} : \mathit{society.members} \mid \\
\quad (\mathit{legaldelegation}\ (\mathit{normativeAg}\ \mathit{self}, \mathit{normativeAg}\ \mathit{ag}, g, \mathit{society}) \lor \\
\quad \mathit{punishdelegation}\ (\mathit{normativeAg}\ \mathit{self}, \mathit{normativeAg}\ \mathit{ag}, g, \mathit{societystate}) \lor \\
\quad \mathit{rewarddelegation}\ (\mathit{normativeAg}\ \mathit{self}, \mathit{normativeAg}\ \mathit{ag}, g, \mathit{societystate}) \lor \\
\quad \mathit{benefiteddelegation}\ (\mathit{normativeAg}\ \mathit{self}, \mathit{normativeAg}\ \mathit{ag}, g, \mathit{society}))\} \\
\hline
\end{array}
$$

Since our categorisation of plans is based on the identification of agents liable to be influenced, we introduce a normative agent which, given a goal, is able to identify potential partners to whom the goal can be delegated, by joining the *CircumstantialPartners* and the *InstitutionalPartners* schemas as follows.

$$\mathit{PPNormativeAgent} \mathrel{\widehat{=}} [\mathit{CircumstantialPartners};\ \mathit{InstitutionalPartners}]$$

6.3.3 Classification of Plans

In this thesis, a specific model of partial plans is used, as described in Section 3.4.5. The basic structure of a plan includes the goal that can be satisfied by executing the plan, the state of the environment that must hold for the plan to be executed, and the plan body that represents the sequence of actions and subgoals that must be satisfied in order to achieve the desired goal. This is a flexible representation of plans since not all the actions are predetermined, and the way to achieve a subgoal is not decided until that subgoal is reached in the original plan. We assume that every agent has a plan library (which contains all the *recipes* for action that the agent knows about), and that all the *actions* included in a plan are among the agent's capabilities, but no restrictions are made regarding *subgoals* included in the plan. By making this assumption, we concentrate our efforts on the delegation of goals rather than on the delegation of actions, because autonomous agents only need to know *what* they have to do rather than *how* to do it.

6.3.3.1 Self-sufficient Plans

The first category of plans corresponds to the *self-sufficient* plans defined by Luck and d'Inverno in [121]. A self-sufficient plan is any plan that involves actions and subgoals than can be satisfied by the agent itself. Formally, a recursive definition is used as follows. The first statement in the definition states that all plan actions must be among the agent's capabilities. The halting condition is reached when the body of the evaluated plan includes just actions, i.e. when the body does not include subgoals (as stated in the second statement). Then, recursion is used to verify that for all subgoals in the plan body there is a self-sufficient plan to satisfy them.

$$\begin{array}{|l}
selfsuffplan\_ : \mathbb{P}(PPNormativeAgent \times Plan) \\
\hline
\forall ppag : PPNormativeAgent;\ p : Plan \bullet \\
\quad selfsuffplan\ (ppag, p) \Leftrightarrow (planactions\ p.body \subseteq ppag.capabilities \land \\
\qquad (plangoals\ p.body = \varnothing \lor \\
\qquad (\forall sg : (plangoals\ p.body) \bullet (\exists p_1 : ppag.planlibrary \bullet \\
\qquad\qquad (sg = p_1.goal \land selfsuffplan\ (ppag, p_1))))))
\end{array}$$

6.3.3.2 Negotiated Plans

A plan must be *negotiated* if there exists a subgoal in its body which cannot be satisfied by the agent itself, and there is another agent that might be influenced to adopt the goal by *exchange delegation* (i.e. the goal can be delegated to a contractor agent). The formalisation of negotiated plans is given as follows. The first statement in the definition states that all plan actions must be among the agent's capabilities. The next predicate states that there is a subgoal that cannot be satisfied by the agent because it does not have a self-sufficient plan, but can be delegated to a contractor agent.

$$\begin{array}{|l}
negotiatedplan\_ : \mathbb{P}(PPNormativeAgent \times Plan) \\
\hline
\forall ppag : PPNormativeAgent;\ p : Plan \bullet \\
\quad negotiatedplan\ (ppag, p) \Leftrightarrow (planactions\ p.body \subseteq ppag.capabilities \land \\
\qquad (\exists sg : (plangoals\ p.body) \bullet \\
\qquad\quad (\neg\ (\exists p_1 : ppag.planlibrary \bullet \\
\qquad\qquad (sg = p_1.goal \land selfsuffplan\ (ppag, p_1))) \land \\
\qquad\quad ppag.contractorags\ (sg) \neq \varnothing)))
\end{array}$$

6.3.3.3 Supported Plans

A plan can be executed with the cooperation of supportive agents if all subgoals included in the plan's body can either be satisfied by the agent itself or delegated to a *captive* agent (i.e. supportive agents or agents from whom reciprocation is expected). These kinds of plans are called *supported plans*, and their formalisation is given below. The first statement in the definition states that all plan actions must be among the agent's capabilities. The second predicate states that for all subgoals, in the plan body, either there is a self-sufficient plan to satisfy it, or the goal can be delegated to a captive agent.

$$\begin{array}{l}
\mathit{supportedplan\_} : \mathbb{P}(\mathit{PPNormativeAgent} \times \mathit{Plan}) \\
\hline
\forall \mathit{ppag} : \mathit{PPNormativeAgent};\ p : \mathit{Plan} \bullet \\
\quad \mathit{supportedplan}\ (\mathit{ppag}, p) \Leftrightarrow (\mathit{planactions}\ p.\mathit{body} \subseteq \mathit{ppag}.\mathit{capabilities} \land \\
\qquad (\forall \mathit{sg} : (\mathit{plangoals}\ p.\mathit{body}) \bullet \\
\qquad\quad ((\exists p_1 : \mathit{ppag}.\mathit{planlibrary} \bullet \\
\qquad\qquad (\mathit{sg} = p_1.\mathit{goal} \land \mathit{selfsuffplan}\ (\mathit{ppag}, p_1))) \lor \\
\qquad\quad (\mathit{ppag}.\mathit{captiveags}(\mathit{sg}) \neq \varnothing))))
\end{array}$$

6.3.3.4 Social Supported Plans

Social supported plans are those that can be performed with the cooperation of agents that are members of the same society and addressees of norms that specify their responsibilities. That is, a social supported plan is one whose subgoals can either be satisfied by the agent itself, or delegated to a member of the society. The formal representation of this is as follows. The first statement in the definition states that all plan actions must be among the agent's capabilities. The second predicate states that for all subgoals, in the plan body, either there is a self-sufficient plan or the goal can be delegated to an institutional partner.

$$\begin{array}{l}
\mathit{socialsupportplan\_} : \mathbb{P}(\mathit{PPNormativeAgent} \times \mathit{Plan}) \\
\hline
\forall \mathit{ppag} : \mathit{PPNormativeAgent};\ p : \mathit{Plan} \bullet \\
\quad \mathit{socialsupportplan}\ (\mathit{ppag}, p) \Leftrightarrow (\mathit{planactions}\ p.\mathit{body} \subseteq \mathit{ppag}.\mathit{capabilities} \land \\
\qquad (\forall \mathit{sg} : (\mathit{plangoals}\ p.\mathit{body}) \bullet \\
\qquad\quad ((\exists p_1 : \mathit{ppag}.\mathit{planlibrary} \bullet \\
\qquad\qquad (\mathit{sg} = p_1.\mathit{goal} \land \mathit{selfsuffplan}\ (\mathit{ppag}, p_1))) \lor \\
\qquad\quad \mathit{ppag}.\mathit{institutionalags}(\mathit{sg}) \neq \varnothing)))
\end{array}$$

6.3.3.5 Feasible Plans

A plan is *feasible* if all subgoals in its body can be achieved either by the agent itself, by one of its supportive agents or by a member of the society. Having a feasible plan means the agent can almost guarantee the satisfaction of the corresponding goal, because, although the agent may not be able to satisfy a subgoal, it has the means to influence other agents to do it. The formalisation of a feasible plan is given below. The first statement in the definition states that all plan actions must be among the agent's capabilities. Then, the next predicate indicates that for all subgoals, in the plan body, either there is a self-sufficient plan to satisfy it, or the goal can be delegated to a captive agent or to an institutional partner.

$$
\begin{array}{|l}
\mathit{feasibleplan\_} : \mathbb{P}(\mathit{PPNormativeAgent} \times \mathit{Plan}) \\
\hline
\forall \mathit{ppag} : \mathit{PPNormativeAgent};\ p : \mathit{Plan} \bullet \\
\quad \mathit{feasibleplan}\ (\mathit{ppag}, p) \Leftrightarrow (\mathit{planactions}\ p.\mathit{body} \subseteq \mathit{ppag}.\mathit{capabilities} \land \\
\qquad (\forall \mathit{sg} : (\mathit{plangoals}\ p.\mathit{body}) \bullet \\
\qquad\quad ((\exists p_1 : \mathit{ppag}.\mathit{planlibrary} \bullet \\
\qquad\qquad (\mathit{sg} = p_1.\mathit{goal} \land \mathit{selfsuffplan}\ (\mathit{ppag}, p_1))) \lor \\
\qquad\quad \mathit{ppag}.\mathit{captiveags}\ (\mathit{sg}) \neq \varnothing \lor \\
\qquad\quad \mathit{ppag}.\mathit{institutionalags}\ (\mathit{sg}) \neq \varnothing)))
\end{array}
$$

6.3.3.6 Aggressive Plans

An *aggressive* plan is a plan whose body includes a subgoal that can only be delegated to a threatened agent. This means, to satisfy the corresponding goal, by using this plan, the agent must threaten another agent to succeed. Formally, it is represented as follows. The first statement in the definition states that all plan actions must be among the agent's capabilities. The predicate states that there is a subgoal, in the plan body, for which a self-sufficient plan does not exist, but there is a threatened agent to which the goal can be delegated.

$$
\begin{array}{|l}
\mathit{aggressiveplan\_} : \mathbb{P}(\mathit{PPNormativeAgent} \times \mathit{Plan}) \\
\hline
\forall \mathit{ppag} : \mathit{PPNormativeAgent};\ p : \mathit{Plan} \bullet \\
\quad \mathit{aggressiveplan}\ (\mathit{ppag}, p) \Leftrightarrow (\mathit{planactions}\ p.\mathit{body} \subseteq \mathit{ppag}.\mathit{capabilities} \land \\
\qquad (\exists \mathit{sg} : (\mathit{plangoals}\ p.\mathit{body}) \bullet \\
\qquad\quad ((\neg\ (\exists p_1 : \mathit{ppag}.\mathit{planlibrary} \bullet \\
\qquad\qquad (\mathit{sg} = p_1.\mathit{goal} \land \mathit{selfsuffplan}\ (\mathit{ppag}, p)))) \land \\
\qquad\quad \mathit{ppag}.\mathit{threatenags}\ (\mathit{sg}) \neq \varnothing)))
\end{array}
$$

6.3.3.7 Risky Plans

Finally, a plan is *risky* if its body includes a subgoal for which there is no agent to whom delegate it. That is, by choosing a plan of this kind, an agent is taking the risk that its goal cannot be satisfied because when a subgoal that can neither be satisfied nor delegated is reached, the plan must be stopped. However, the possibility exists that a new agent, which can be influenced, may enter the society before the subgoal is reached, which means that the subgoal can then be delegated. The formal representation of a risky plan is given below. The first statement in the definition states that all plan actions must be among the agent's capabilities, then, the predicate states that there is a subgoal, in the plan body, for which a self-sufficient plan does not exist, and for which there is neither a contractor, nor a captive agent, nor an institutional partner, nor a threatened agent to delegate it.

$$
\begin{array}{|l}
\mathit{riskyplan\_} : \mathbb{P}(\mathit{PPNormativeAgent} \times \mathit{Plan}) \\
\hline
\forall \mathit{ppag} : \mathit{PPNormativeAgent};\ p : \mathit{Plan} \bullet \\
\quad \mathit{riskyplan}\ (\mathit{ppag}, p) \Leftrightarrow (\mathit{planactions}\ p.\mathit{body} \subseteq \mathit{ppag}.\mathit{capabilities} \land \\
\qquad (\exists\, \mathit{sg} : (\mathit{plangoals}\ p.\mathit{body}) \bullet \\
\qquad\quad ((\neg\ (\exists\, p_1 : \mathit{ppag}.\mathit{planlibrary} \bullet \\
\qquad\qquad (\mathit{sg} = p_1.\mathit{goal} \land \mathit{selfsuffplan}\ (\mathit{ppag}, p)))) \land \\
\qquad\quad \mathit{ppag}.\mathit{contractorags}\ (\mathit{sg}) = \varnothing \land \\
\qquad\quad \mathit{ppag}.\mathit{captiveags}(\mathit{sg}) = \varnothing \land \\
\qquad\quad \mathit{ppag}.\mathit{institutionalags}(\mathit{sg}) = \varnothing \land \\
\qquad\quad \mathit{ppag}.\mathit{threatenags}\ (\mathit{sg}) = \varnothing)))
\end{array}
$$

6.3.4 Discussion

By using these categories of plans, different strategies to select plans can be defined by giving priorities to a range of alternative plans. For example, when a plan is not *self-sufficient*, agents might select as a first option a *supported* plan in order to delegate their goals to those agents from whom support is always obtained. If this is not possible, a *social supported* plan may be chosen to take advantage of the power given by norms to delegate goals. If these two options are not possible, agents can either choose a plan in which negotiation techniques must be used, or a plan for which agents must be threatened to adopt its subgoals.

Note that different kinds of agents can be defined by stating different priorities to select a plan. For example, some agents might prefer a *feasible* plan in the first instance and, if this is not possible, use a *supported* plan, then a *social supported* plan and as last resort an *aggressive* plan. However, other opportunistic agents might prefer in the first instance to use an aggressive plan, then a supported plan, and so on, a self-sufficient plan being their last resort.

6.4 Goal Adoption

6.4.1 Introduction

Goal adoption can be defined as a voluntary process through which an agent decides to satisfy the goal of another based on its goals and motivations [119]. Thus, rather than being forced, autonomous agents must be convinced to adopt external goals. Goal adoption is the counterpart of goal delegation. To delegate a goal, an agent identifies when its powers can be exerted to influence other agents. Conversely, to adopt a goal, agents must recognise the power of delegating agents and how these powers affect their own goals before allowing themselves to be influenced. So, agents are liable to be influenced by any of the following reasons: some of their important goals can be hindered; some goals can be exchanged; they are supportive agents; they must reciprocate the others' past actions; they are under the authority of others; they must be penalised; they must give rewards to compliant agents; or they are addressees of a norm that benefits others. As can be seen, these reasons correspond to the situations in which powers can be used to delegate goals as described in Section 6.2. In the remainder of this section, we provide the means for agents to identify when the circumstantial and institutional powers of other agents must be recognised to allow themselves to be *influenced* to adopt external goals.

6.4.2 Recognising Circumstantial Powers

6.4.2.1 Threatened Adoption

A threatened agent adopts the external goals of another if it believes that the other agent can hinder more important goals than those that could be hindered otherwise. Formally, an agent adopts external goals if the delegating agent has *illegal coercive* power regarding one of the agent's goals that is more important than the goals hindered by external goals. This is represented in the *ThreatenAdoption* schema where the schema of a normative agent is included, together with two variables: one to represent the delegating agent (*delegatingag*) and the other to represent the external goals (*externalgs*). The first part of the predicate states that the delegating agent must have illegal coercive power regarding a goal g, whereas the second part makes clear that this goal is more important than those hindered by the external goals.

ThreatenAdoption

$$
\begin{array}{l}
\mathit{NormativeAgent} \\
\mathit{delegatingag} : \mathit{AgentName} \\
\mathit{externalgs} : \mathbb{P}\, \mathit{Goal} \\
\hline
\exists\, g : \mathit{goals} \bullet \\
\quad (\mathit{illegalcoercivepower}\ (\mathit{normativeAg}\ \mathit{delegatingag},\ \mathit{normativeAg}\ \mathit{self}, \\
\qquad\qquad\qquad\qquad\qquad\qquad\qquad g,\ \mathit{beliefs}) \land \\
\quad (\mathit{importance}\ (\mathit{gms},\ \{g\}) > \\
\qquad \mathit{importance}\ (\mathit{gms},\ \mathit{hindered}\ (\mathit{goals}, \mathit{externalgs}))))
\end{array}
$$

6.4.2.2 Exchange adoption

Agents also can agree to adopt external goals via exchange of goals if these goals do not hinder goals that are more important than the goals to be exchanged. The *ExchangeAdoption* schema represents agents that are influenced through exchange of goals. The declaration part of the schema is similar to the declaration part of *ThreatenAdoption* schema except that, here, a variable to represent the goals to be exchanged (*exchangedgs*) is included. The first predicate indicates that the set of goals to be exchanged belong to the agent. The second predicate states that the delegating agent must have exchange power over this agent regarding the external and exchanged goals. The third predicate represents the agent acting according to its motivations, i.e. the importance of the exchanged goals must be higher than the importance of the goals being hindered by the external goals.

ExchangeAdoption

$$
\begin{array}{l}
\mathit{NormativeAgent} \\
\mathit{delegatingag} : \mathit{AgentName} \\
\mathit{externalgs} : \mathbb{P}\, \mathit{Goal} \\
\mathit{exchangedgs} : \mathbb{P}\, \mathit{Goal} \\
\hline
\mathit{exchangedgs} \subseteq \mathit{goals} \\
\forall\, \mathit{excg} : \mathit{exchangedgs} \bullet (\exists\, \mathit{extg} : \mathit{externalgs} \bullet \\
\quad \mathit{exchangepower}\ (\mathit{normativeAg}\ \mathit{delegatingag}, \mathit{extg}, \mathit{normativeAg}\ \mathit{self}, \\
\qquad\qquad\qquad\qquad\qquad\qquad\qquad \mathit{excg},\ \mathit{beliefs})) \\
(\mathit{importance}\ (\mathit{gms},\ \mathit{exchangedgs}) > \\
\qquad \mathit{importance}\ (\mathit{gms},\ \mathit{hindered}\ (\mathit{goals},\ \mathit{externalgs})))
\end{array}
$$

6.4.2.3 Supportive Adoption

Goals can also be adopted as an end. However, autonomous agents do not do this for every agent that requires goal adoption but only for those delegating agents that belong to a group of special agents. We have called them supportive agents because in the same way that an agent may adopt a goal (of any one in the group) as an end, it can also delegate a goal to anyone. To express this, the *SupportiveAdoption* schema is introduced where variables to represent the delegating agent (*delegatingag*), the external goals (*externalgs*), and the group of supportive agents (*supportiveags*), are included. The first predicate indicates that to be satisfied, the external goals must be within the goals of the delegating agent. The second predicate states that the agent must believe the delegating agent has support power regarding the external goals.

$$
\begin{array}{|l}
\multicolumn{1}{l}{\textit{SupportiveAdoption}} \\
\hline
\textit{NormativeAgent} \\
\textit{delegatingag} : \textit{AgentName} \\
\textit{externalgs} : \mathbb{P}\,\textit{Goal} \\
\textit{supportiveags} : \mathbb{P}\,\textit{AgentName} \\
\hline
\textit{externalgs} \subseteq (\textit{normativeAg delegatingag}).\textit{goals} \\
\forall g : \textit{externalgs} \bullet \textit{supportpower}\ (\textit{normativeAg delegatingag}, \\
\qquad\qquad\qquad \textit{normativeAg self}, \textit{supportiveags}, g, \textit{beliefs}) \\
\hline
\end{array}
$$

It might also be that benevolent adoption of external goals is agreed only if no goals are hindered by providing this help. This is formally represented in the *ConditionedAdoption* schema which is similar to the schema above, but with an extra condition in the third predicate. It states that no goal is hindered by any of the adopted external goals.

$$
\begin{array}{|l}
\multicolumn{1}{l}{\textit{ConditionedAdoption}} \\
\hline
\textit{NormativeAgent} \\
\textit{delegatingag} : \textit{AgentName} \\
\textit{externalgs} : \mathbb{P}\,\textit{Goal} \\
\textit{supportiveags} : \mathbb{P}\,\textit{AgentName} \\
\hline
\textit{externalgs} \subseteq (\textit{normativeAg delegatingag}).\textit{goals} \\
\forall g : \textit{externalgs} \bullet \textit{supportpower}\ (\textit{normativeAg delegatingag}, \\
\qquad\qquad\qquad \textit{normativeAg self}, \textit{supportiveags}, g, \textit{beliefs}) \\
\textit{hindered}\ (\textit{goals}, \textit{externalgs}) = \varnothing \\
\hline
\end{array}
$$

We can go even further and model the case in which external goals of friends are adopted only if these goals do not hinder goals whose importance is above a certain value, represented by the *limit* variable, as follows.

$$
\begin{array}{|l}
\multicolumn{1}{l}{\mathit{LimitedAdoption}} \\
\hline
\mathit{NormativeAgent} \\
\mathit{delegatingag} : \mathit{AgentName} \\
\mathit{externalgs} : \mathbb{P}\, \mathit{Goal} \\
\mathit{supportiveags} : \mathbb{P}\, \mathit{AgentName} \\
\mathit{limit} : \mathbb{N} \\
\hline
\mathit{externalgs} \subseteq (\mathit{normativeAg}\ \mathit{delegatingag}).\mathit{goals} \\
\forall g : \mathit{externalgs} \bullet \mathit{supportpower}\ (\mathit{normativeAg}\ \mathit{delegatingag}, \\
\qquad\qquad\qquad \mathit{normativeAg}\ \mathit{self},\ \mathit{supportiveags},\ g,\ \mathit{beliefs}) \\
\mathit{importance}\ (\mathit{gms},\ \mathit{hindered}\ (\mathit{goals},\ \mathit{externalgs})) \leq \mathit{limit} \\
\hline
\end{array}
$$

6.4.2.4 Reciprocate Adoption

Now, agents adopting external goals to reciprocate other agents' past actions must believe that the delegating agent is an agent that has previously complied with a norm and thus requires reciprocation. This is fomalised in the *ReciprocateAdoption* schema, which includes the *SocietyNormativeAgent* schema that represents a normative agent with models of both its society (*society*) and the state of this society (*societystate*). Variables to represent the delegating agent (*delegatingag*), and the external goal (*externalgs*) are also included. The predicate states that the delegating agent must have reciprocation power over this agent.

$$
\begin{array}{|l}
\multicolumn{1}{l}{\mathit{ReciprocateAdoption}} \\
\hline
\mathit{SocietyNormativeAgent} \\
\mathit{delegatingag} : \mathit{AgentName} \\
\mathit{externalgs} : \mathbb{P}\, \mathit{Goal} \\
\hline
\mathit{reciprocationpower}(\mathit{normativeAg}\ \mathit{delegatingag},\ \mathit{normativeAg}\ \mathit{self}, \\
\qquad\qquad\qquad \mathit{societystate},\ \mathit{beliefs}) \\
\hline
\end{array}
$$

We can also model the case in which an agent reciprocates only if no current goals are hindered by the external goals. To do so we use the *ConditionedReciprocateAdoption* schema that is almost the same that *ReciprocateAdoption* schema except that now an additional constraint is added. This constraint states that no goals must be hindered by the external goals.

$$
\begin{array}{|l}
\multicolumn{1}{l}{\_\ \mathit{ConditionedReciprocateAdoption}\ \_\_\_\_\_\_\_\_} \\
\mathit{SocietyNormativeAgent} \\
\mathit{delegatingag} : \mathit{AgentName} \\
\mathit{externalgs} : \mathbb{P}\, \mathit{Goal} \\
\hline
\mathit{reciprocationpower}(\mathit{normativeAg}\ \mathit{delegatingag},\ \mathit{normativeAg}\ \mathit{self}, \\
\qquad\qquad\qquad\qquad\qquad\qquad \mathit{societystate},\ \mathit{beliefs}) \\
\mathit{hindered}\ (\mathit{goals},\ \mathit{externalgs}) = \varnothing \\
\hline
\end{array}
$$

Similarly, agents can reciprocate other agents' past actions by adopting some of their goals only in those cases in which the external goals do not hinder goals whose importance is above a certain limit. The formal representation of this is given in the *LimitedReciprocateAdoption* schema, where the predicates state that the delegating agent must have reciprocation power, and that the external goals do not hinder goals whose importance is above a certain limit.

$$
\begin{array}{|l}
\multicolumn{1}{l}{\_\ \mathit{LimitedReciprocateAdoption}\ \_\_\_\_\_\_\_\_} \\
\mathit{SocietyNormativeAgent} \\
\mathit{delegatingag} : \mathit{AgentName} \\
\mathit{externalgs} : \mathbb{P}\, \mathit{Goal} \\
\mathit{limit} : \mathbb{N} \\
\hline
\mathit{reciprocationpower}(\mathit{normativeAg}\ \mathit{delegatingag},\ \mathit{normativeAg}\ \mathit{self}, \\
\qquad\qquad\qquad\qquad\qquad\qquad \mathit{societystate},\ \mathit{beliefs}) \\
\mathit{importance}\ (\mathit{gms},\ \mathit{hindered}\ (\mathit{goals},\ \mathit{externalgs})) \leq \mathit{limit} \\
\hline
\end{array}
$$

These kinds of goal adoption can explain how some groups are created. If an agent rejects the adoption of one group member's goals or does not reciprocate, others can eliminate this agent from the group.

6.4.3 Recognising Institutional Powers

6.4.3.1 Subordinate Adoption

Once an agent has decided to remain in a society, it also accepts the designated authorities and their orders. As discussed in Subsection 5.4.4, agents remain in a society due to two different kinds of reasons.

- The society contributes through the responsibilities of its members, to the satisfaction of goals that are more important than those goals hindered by agent responsibilities (we have called this the *social satisfaction* condition).

- Agents are involved in relationships with others that cause them to be tied to the society.

Both reasons have already been represented in the *StayingbyGoals* and the *StayingbyTies* formalisations given in Subsection 5.4.4. Here, we join them, through a disjunction operation, to represent an agent that has decided to remain in a society because its goals are being satisfied or because it has certain relationships with other agents.

$$StayinginSociety \mathrel{\widehat{=}} StayingbyGoals \lor StayingbyTies$$

Now, the *SubordinateAdoption* schema represents the adoption of external goals which are delegated by the authority of a society in which the agent participates. The *StayinginSociety* definition is included to make clear that this adoption takes place as long as the agent has reasons to continue participating in the society (*society*). Variables to represent the delegating agent (*delegating*) and the external goal (*externalgs*) are also included. The predicate states that the delegating agent must have legal power over the agent in the considered society, i.e. it must be an authority.

$$\begin{array}{|l}
\text{\_\_}\mathit{SubordinateAdoption}\text{\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_} \\
\mathit{StayinginSociety} \\
\mathit{delegatingag} : \mathit{AgentName} \\
\mathit{externalgs} : \mathbb{P}\,\mathit{Goal} \\
\hline
\mathit{legalpower}\ (\mathit{normativeAg}\ \mathit{delegatingag}, \mathit{normativeAg}\ \mathit{self}, \mathit{society}) \\
\hline
\end{array}$$

No orders issued by an authority can be ignored by the members of a society, at least until these agents emigrate to another society and, therefore, they are beyond the authority's domain.

6.4.3.2 Punished Adoption

A normative agent must be responsible for its actions and, consequently, it must accept the punishments it deserves from having violated a norm. However, these punishments are only applied by agents entitled to punish the violated norm, i.e. those that have legal punishment power. This authority, as part of its responsibilities, may issue orders (goals) that offenders of norms must accept (adopt). Since the power of authorities is constrained, for agents to adopt a goal they must be sure that the goal corresponds to the punishments associated with the violated norm. This avoids agents being abused by empowered agents.

Formally, the conditions for agents to adopt external goals by accepting punishments are listed as follows.

- The agent must be interested in staying in the society.
- The delegating agent must have legal punishment power over the agent to penalise a norm n. This means that delegating agents are entitled to punish a norm n and the agent is an addressee that has violated the norm n.
- The external goal must correspond to the punishments of the norm n.

The *PunishedAdoption* schema formalises these conditions. There, the *StayinginSociety* definition is included to represent the first condition. Variables to represent the delegating agent (*delegating*) and the external goals (*externalgs*) are also included. The two last mentioned conditions are represented in the unique predicate of the schema.

$$
\begin{array}{l}
\textit{PunishedAdoption} \\
\hline
\textit{StayinginSociety} \\
\textit{delegatingag} : \textit{AgentName} \\
\textit{externalgs} : \mathbb{P}\, \textit{Goal} \\
\hline
\exists\, n : \textit{norms} \bullet \\
\quad (\textit{legalpunishmentpower}\ (\textit{normativeAg}\ \textit{delegatingag}, \textit{normativeAg}\ \textit{self}, \\
\qquad\qquad\qquad\qquad\qquad\qquad n, \textit{societystate}) \land \\
\quad \textit{externalgs} \subseteq n.\textit{punishments}) \\
\hline
\end{array}
$$

6.4.3.3 Rewarded Adoption

External goals are also adopted if these goals correspond to the rewards offered for having fulfilled a norm. The process is similar to the adoption of goals due to punishments. Thus, the conditions to adopt external goals by rewarding compliant agents are as follows.

- The agent must be interested in staying in the society.
- The agent believes that the delegating agent has legal reward power regarding a norm n.

- The external goals must be in the rewards of the norm n.

The *RewardingAdoption* schema formalises these conditions. The *StayinginSociety* schema is included to represent the first condition. Variables to represent the delegating agent (*delegating*) and the external goals (*externalgs*) are also included. The two last mentioned conditions are represented in the predicate.

$$
\begin{array}{|l}
\textit{RewardingAdoption} \\
\hline
\textit{StayinginSociety} \\
\textit{delegatingag} : \textit{AgentName} \\
\textit{externalgs} : \mathbb{P}\, \textit{Goal} \\
\hline
\exists\, n : \textit{norms} \bullet \\
\quad (\textit{legalrewardpower}\ (\textit{normativeAg}\ \textit{delegatingag}, \textit{normativeAg}\ \textit{self}, \\
\qquad\qquad\qquad\qquad\qquad\qquad n, \textit{societystate}) \land \\
\quad (\textit{externalgs} \subseteq n.\textit{rewards})) \\
\hline
\end{array}
$$

6.4.3.4 Benefited Adoption

All agents that have adopted a norm have the responsibility to satisfy its normative goals. Consequently, any external goals that are included in these normative goals, and are delegated by a beneficiary of the norm, must be adopted as a way of recognising the legal benefit power of the agent. Formally, external goals are adopted if the delegating agent has legal benefit power regarding a norm whose normative goals include the external goals. This is represented in the schema *BenefitedAdoption*. The *StayinginSociety* definition is included to state that this adoption takes place as long as the agent continues participating in the society (*society*). Variables to represent the delegating agent (*delegating*) and the external goal (*externalgs*) are also included. The predicate states that the delegating agent must have legal benefit power over the agent and that external goals are part of the normative goals of the norm causing the power.

$BenefitedAdoption$

$$StayinginSociety$$
$$delegatingag : AgentName$$
$$externalgs : \mathbb{P}\, Goal$$

$$\begin{array}{l}\exists\, n : norms \bullet \\ \quad (legalbenefitpower\ (normativeAg\ delegatingag, normativeAg\ self, \\ \qquad\qquad\qquad\qquad\qquad\qquad n, society) \wedge \\ \quad (externalgs \subseteq n.normativegoals))\end{array}$$

6.4.4 Discussion

In Table 6.2, the conditions that must prevail for an agent to adopt goals delegated by an empowered agent are summarised. In the table, $Agent_D$ is the agent delegating goals, $Agent_A$ is the agent adopting goals, *externalgs* are the goals being delegated, *hinderedgs* are the goals of $Agent_A$ that could be hindered by satisfying the external goals, *threatenedgs* are the $Agent_A$'s goals that can be hindered by exerting illegal coercive power, and *exchangegs* are the $Agent_A$'s goals to be exchanged.

Type of Adoption	$Agent_D$'s powers	$Agent_A$'s states
Threatened Adoption	Illegal Coercive Power	$Importance(threatenedgs) \geq Importance(hinderedgs)$
Exchange Adoption	Exchange Power	$Importance(exchangegs) \geq Importance(hinderedgs)$
Benevolent Adoption	Support Power	–
Conditioned Adoption	Support Power	$hinderedgs = \varnothing$
Limited Adoption	Support Power	$Importance(hinderedgs) \geq Limit$
Reciprocated Adoption	Reciprocation Power	–
Reciprocated Conditioned Adoption	Reciprocation Power	$hinderedgs = \varnothing$
Reciprocated Limited Adoption	Reciprocation Power	$Importance(hinderedgs) \geq Limit$
Obeyed Adoption	Legal Power	Membership
Punished Adoption	Legal Punish Power	Offender of a norm
Rewarding Adoption	Legal Reward Power	Promoter of a norm
Benefited Adoption	Legal Benefit Power	Addressee of a norm

TABLE 6.2: Adoption of Goals by Recognising Agent Powers

Since the adoption of external goals implies *commitments* towards other agents, the non satisfaction of these goals can be penalised, and their complete satisfaction can be rewarded. We argue that the adoption of external goals must be made *formal* through the adoption of norms whose normative goals correspond to these external goals, whose addressees correspond to the agents adopting the goals and, sometimes, whose issuer coincides with the agent delegating the goals. These norms might already exist, such as the obligations of agents in an organisation to help their colleagues, or they can be created at the point at which agents agree to adopt the external goals. Thus, new norms can be adopted by agents at run-time and not only at the point when agents *join* a society as explained in Section 5.4.2. As can be seen, norm adoption is a decision-making process very related to adoption of goals. The norm adoption problem is faced in the next section.

6.5 Autonomous Norm Adoption

6.5.1 Introduction

Introducing agents able to adopt new norms is an important step towards the representation of dynamic societies where changes in current legislation might occur, the society members are not necessarily predetermined, and where relationships between members are created and destroyed dynamically. Enabling agents to adopt new norms allows both the independent design of these agents (because they do not need prior knowledge of the norms they must fulfill), and the possibility for agents to join or leave a society without changing their internal design. In addition, since norms represent the responsibilities of agents, and norms are different in each society, agents become able to adopt different *roles* and obligations [105]. Moreover, the ability to adopt norms enables agents to make agreements with other agents at run-time, to either adopt or delegate their goals.

Despite its importance, the process of norm adoption has received little attention from the agent research community, in part due to norms initially being considered as built-in constraints [17, 127, 153] where the issuing and adoption of new norms is not considered. Recent approaches admit that new norms can be issued; however, they assume that agents will adopt any new norm issued by a recognised authority [13, 69] without considering the reasons that such agents might have to do so. Our main contribution to this area is the statement that autonomous agents adopt a new norm when its issuer is recognised as someone with power that can be exerted. Notice that we are distinguishing *having* power from *exercising* power because we recognise that power is constrained. Being able to understand this difference allows agents to recognise exploitation or abusive situations when empowered agents' orders stretch beyond their domain of competence (i.e. when the scope of their power is exceeded).

6.5.2 The Norm Adoption Process

Norm adoption can be better defined as the process through which agents recognise their responsibilities towards other agents by internalising the norms that specify these responsibilities. The importance of *norm adoption* as a voluntary process has been already pointed out by many [29, 38, 43]. Their research, rather than explaining why norms are adopted, describes the cases in which norms must be rejected. These include situations in which the issuer is not an authority; the norms are not within the competence of an authority; addressees are not under the authority's domain; the context in which norms are issued is not appropriate; norms are issued to satisfy an authority's personal interest; or norms are not intended to be beneficial for the group.

Although these causes of rejection are important, they cannot be taken as general conditions to reject norms because they are neither applicable to all kinds of norms nor in all situations. The concept of *authority* refers to the power assigned according to norms and accepted as legitimate by all members of a society. So, adopting only those norms issued by an authority covers the case of norms issued in a very well defined social structure. However, as discussed in earlier chapters of this thesis, there are other kinds of norms which are not necessarily issued by a recognised authority, such as those norms that result from agreements between agents, and those that emerge from customs of behaviour.

Now, recognising when norms are issued to satisfy the personal interests of the issuer is not an easy task and, although this might be important for societies in which the primary objective is the equality of the members, it is too restrictive for other kinds of societies or groups. For example, suppose that a businessman wants to create a private enterprise, and one of his goals is be to obtain profits. The majority of the enterprise's norms will be issued in order to guarantee the achievement of this goal, and although the norms represent the businessman's interests, employees adopt them, and as long as they want to remain in the organisation the norms issued by the businessman will be adopted. Similarly, members of a gang of armed robbers might recognise the gang leader as someone with enough power to issue orders although the goods they get from their felonies are not equally distributed among them. This suggests that the motives for issuing a norm do not always coincide with the motives for adopting the norm, and a balance of interests must exist between issuers and addressees of a norm.

We state that for a norm to be adopted, three conditions must be satisfied:

- agents must recognise themselves as addressees of the norm;
- the norm must not be already adopted; and

- the norm must have been issued by an empowered agent.

The fist two conditions are considered in the general process for a normative agent to adopt a norm which is formally represented in the *NormAdoption* schema. The third condition changes depending on the kind of power possessed by the issuer. So, it will be included later on, when the specific reasons to adopt a norm are explained. At the moment only the general model of norm adoption is given. The *NormAdoption* schema includes the *newnorm*? variable that represents the norm waiting to be adopted (as an input), and the *issuer* variable that represents the agent issuer of the norm. The first predicate states that the norm must be directed to the agent. The second predicate makes clear that the new norm must not be part of the set of already adopted norms. The third predicate relates the new norm and its issuer. Finally, the last predicate represents the adoption of the new norm by the agent.

$$
\begin{array}{|l}
\textit{NormAdoption} \\
\hline
\Delta \textit{NormativeAgent} \\
\textit{newnorm}? : \textit{Norm} \\
\textit{issuer} : \textit{AgentName} \\
\hline
\textit{self} \in \textit{newnorm}?.\textit{addressees} \\
\textit{newnorm}? \notin \textit{norms} \\
\textit{issuedby}\ (\textit{newnorm}?,\ \textit{issuer}) \\
\textit{norms}' = \textit{norms} \cup \{\textit{newnorm}?\} \\
\hline
\end{array}
$$

6.5.3 Reasons to Adopt New Norms

On the one hand, we have stated that norm adoption must be the formal part of the acceptance of external goals. On the other, we have argued that to adopt a norm, its issuer must be recognised as an agent with power. This latter is not a contradiction with our first argument since to adopt external goals, agents recognise the power of delegating agents, and complying with norms means satisfying goals whose benefits may be enjoyed by another. Consequently, the reasons to adopt norms must correspond to the reasons to adopt external goals. That is, an agent adopts a norm when it is threatened to do it; when the norm represents a commitment to exchange goals; when a supportive agent may benefit from the norm; when an agent that provided help in the past may benefit from the norm; when the norm has been issued by an agent with authority to legislate or to punish the agent; when the norm corresponds to its responsibilities to reward another agent; or when the normative goals are part of some responsibilities already acquired by the agent. Then, to make goal adoption correspond to norm adoption, we group the reasons agents have to adopt external goals, which correspond to those defined in Section 6.4, as follows.

$$\begin{array}{rl} \mathit{ReasonsToAdopt} \mathrel{\widehat{=}} & \mathit{ThreatenAdoption} \lor \mathit{ExchangeAdoption} \lor \\ & \mathit{SupportiveAdoption} \lor \mathit{ReciprocateAdoption} \lor \\ & \mathit{SubordinateAdoption} \lor \mathit{PunishedAdoption} \lor \\ & \mathit{RewardingAdoption} \lor \mathit{BenefitedAdoption} \end{array}$$

Now, we define the *autonomous* adoption of norms as a process in which agents decide, based on their own goals and motivations, which norms to adopt. The formal representation of this process is given in the *AutonomousNormAdoption* schema where two important schemas are included. The *ReasonToAdopt* schema represents the conditions that must be satisfied to adopt external goals which are delegated by an agent. The *NormAdoption* schema represents the operation to adopt a new norm. To link these, some constraints are given as follows. The first predicate states that the delegating agent must correspond to the issuer of the new norm. The second predicates states that the agent is an addressee of the norm. The third predicate makes clear that the external adopted goals must correspond to the normative goals of the new norm.

$$\begin{array}{|l} \hline \mathit{AutonomousNormAdoption} \\ \mathit{ReasonsToAdopt} \\ \mathit{NormAdoption} \\ \hline \mathit{delegatingag} = \mathit{issuer} \\ \mathit{self} \in \mathit{newnorm?.addressees} \\ \mathit{externalgs} \subseteq \mathit{newnorm?.normativegoals} \\ \hline \end{array}$$

As can be seen, powers are the key to understanding why, although autonomous, agents may still adopt and comply with norm.

6.6 Conclusion

In this chapter, we have shown how *agent powers* affect processes of decision-making such as the delegation of goals, the adoption of goals, and the process to select a plan to achieve a goal. All of this starts with the assumption that the benevolent adoption of goals cannot be guaranteed in a world of autonomous agents and, therefore, mechanisms to influence agents are needed. The key idea is that an agent's powers can be used to *influence* those agents over whom these powers can be exerted.

Social influence has been recognised by many [124, 136, 137] who have also given some of the reasons to delegate and adopt goals described in this chapter. However, previous research takes for granted that agents are able to identify social influence. That is, it does not provide the means for agents to identify situations in which they are liable to be influenced as described in this chapter. In our model, every situation in which agents are influenced is justified not only in terms of how the goals of an agent are affected, but also in terms of the preferences, given by motivations, that agents have over these goals. This is consistent with the principle that autonomous agents take decisions on the basis of their own goals and motivations.

To delegate a goal, an agent must believe it has power over the agent to whom the goal is delegated. In addition, to adopt a goal, an agent must recognise the power of the agent that is delegating the goal. When both situations coincide, an agreement to cooperate can be made. Thus, by exerting powers and by recognising those powers, agents are *influenced* to adopt goals. In this chapter, eight cases in which agents can influence others are considered in the delegation and adoption of goals. Agents can be influenced because: some of their important goals can be hindered; they can exchange some of their goals; they belong to a group of supportive agents; they must reciprocate the actions of other agents; they are under the authority of other agents; they must be penalised; they must provide rewards; or they must comply with their responsibilities in a society.

Notice that powers are always interpreted according to the beliefs of the reasoning agent. That is, on the one hand to *delegate* a goal, an agent must believe it has power over the chosen agent. On the other, to *adopt* an external goal, an agent must believe that the delegating agent has power over it. This means that an agreement can never be achieved if the beliefs of both agents do not coincide. It is also important to say that adoption of goals requires normative agents to be able to recognise and reason about norms because it makes no sense to delegate goals to agents that do not recognise their commitments towards others.

In this chapter, the means for agents to classify other agents as potential agents to delegate goals to are also described. In this way, agents delegating goals identify agents that can be threatened, agents that can exchange goals, agents that can provide support, and agents that help as part of their responsibilities in a society. This classification helps agents to identify beforehand how other agents may react to a petition of goal adoption and, consequently, it can be used to select a plan. According to this classification of agents, a classification of plans is also provided. Seven different kinds of plans are identified. There are plans that can be executed by the agent itself, plans whose subgoals require negotiation with other agents, plans that can be executed with the help of supportive agents, and plans that are executed with the help of other members of a society. In addition, there are plans whose subgoals can all be delegated, plans whose subgoals can only be delegated if other agents are coerced, and plans that include subgoals that cannot be delegated to any agent. This classification of plans provides alternatives for the selection of a plan to satisfy a goal.

In our model, the adoption of norms is described as a process through which an agent shows its willingness to comply with a norm by making an internalisation of it. However, adoption of norms is never taken as an end but as a decision based on an agent's goals and motivations. Since compliance with norms implies the satisfaction of normative goals whose benefits may be enjoyed by others, norm adoption is seen in this chapter as a way to make formal the adoption of external goals. Then, the reasons to adopt external goals are also valid as reasons to adopt new norms. Moreover, for a norm to be adopted the issuer must be recognised as an empowered agent, and the situations for this power to be exerted must hold. In this way, the limits of an agent's power are established, and abusive situations can be avoided. By contrast with other proposals in which norm adoption is also seen as an autonomous decision [38, 43], our model, besides being based on a well defined framework of norms, is more general because it allows the adoption of any kind of norm and not just the adoption of those norms issued in a very well structured institution.

In this chapter we have shown the way powers due to the capabilities of agents and powers given by the norms in a society impact the behaviour of agents. On the one hand, powers are used to influence other autonomous agents to make them satisfy a goal. On the other, powers are used to understand the reason for agents to allow themselves to be influenced to adopt external goals. Powers are the key to effectively satisfying an agent's goals with the help of other agents and, at the same time, they explain why *autonomous* agents adopt and comply with norms. These are the kinds of agents able to exist in societies regulated by norms.

Chapter 7

Autonomous Norm Compliance

7.1 Introduction

A way to understand a society is by seeing it as a social system which, besides social and communication structures, includes a system of norms to control the behaviour of its members [71]. The effectiveness of norms depends on *compliance* with them by all individuals to whom norms are3 addressed. If agents do not comply, social systems become unpredictable, conflicts of interest might appear and, the performance of each agent could be degraded. For a society to be stable, individuals who never dismiss a norm might be preferred. For these reasons, when norms were introduced to the field of multi-agent systems [127, 153], agents that always comply with norms were considered. Now, with agents working to satisfy their own goals but willing to join societies and work for other agents, expecting that some of their goals can be satisfied by doing so, a different perspective for norm compliance must be taken. From the point of view of autonomous agents with their own goals to satisfy, the advantages of norm compliance might not be so obvious, especially when norms clash with their goals. Furthermore, since functioning with norms requires extra computational effort, there are no reasons why an agent should comply with them, yet agents join societies and fulfill their norms. To explain the reasons for autonomous agents to comply with norms and to give models of how these reasoning about norms can be done, are the main concerns of this chapter.

In agent research, compliance with norms has been considered from two different perspectives: agents that always obey norms [13, 108, 153] and agents that autonomously choose whether to do so [9, 12, 29, 54, 116]. Both possibilities may cause conflicts between a society and the individuals within it. Whereas agents that always comply with norms are important for the design of societies in which total control is needed [13, 99, 127], agents that can decide on the basis of their own goals and motivations whether to comply with them are important for the design of dynamic systems in which agents act on behalf of different users and, while satisfying their own goals, are able to join a society and cooperate with other agents. Autonomous norm decision is important to face those situations in which an agent's goals conflict with the norms that control its behaviour inside a society. Agents that deliberate about norms are also needed in systems in which unforseen events might occur. For example, suppose a lawnmower agent whose norms require it to avoid lakes can only avoid hitting a child by veering towards a lake. Clearly, the goal of avoiding the child is more important than complying with the norm directed at preserving an agent's physical structure. Deliberation about norms is also needed in such situations in which agents face conflicting norms, and they have to decide which norm is more important to fulfill. As can be seen, violation of norms is, sometimes, justified.

The importance of modelling compliance with norms as an autonomous agent's decision has been pointed out by several researchers [29, 43, 44, 45], and has been partly addressed by others. Proposals for norm compliance have generally relied on specific decision-making strategies based on how much an agent gains or loses by complying [9, 54], and on the probability of being caught by a defender of a norm [12]. These cases are very specific and, therefore, inadequate to model different kinds of normative behaviour of autonomous agents. A general model of norm compliance by autonomous agents is the objective of this chapter, and to achieve this objective we argue that two important questions must be answered. First, it is necessary to explain what might motivate an agent to dismiss or complying with a norm. Second, whatever decision an agent takes, the way in which this decision affects the goals of an agent must be explained.

Our proposal states that *autonomous norm compliance* involves two processes, one to deliberate about whether to comply with a norm (*the norm deliberation process*), and the other to update the goals, and therefore the intentions of agents accordingly (*the norm compliance process*). Both processes must take into account not only the goals of agents, but also the mechanisms that the society has to avoid violation of norms such as rewards and punishments, that is, agents must consider the so called *social pressure of norms* before making any decision regarding norms.

In norm deliberation, agents employ different *strategies* to make a decision regarding norms. We divide these strategies into three classes: *simple*, *motivated*, and *influenced* strategies for norm-compliance.

- In *simple* norm-compliance strategies, agents do not undertake any reflection at all and only few elements of norms are considered.

- *Motivated* norm-compliance strategies consider the possible effects on an agent's goals of both satisfying the normative goals and acquiring rewards in the case the norm is fulfilled, or satisfying the goals associated with punishments if it is not.

- By contrast, *influenced* norm-compliance strategies consider the normative behaviour of other agents that coexist in the same system to decide what to do next.

Once agents take a decision about which norms to fulfill, *a process of norm compliance* must be started in order to update an agent's goals accordingly. That is, goals and intentions of agents may change due to the following reasons.

- Additional to their current goals, both normative goals (from norms to be fulfilled) and goals associated with the punishment of unfulfilled norms must be satisfied.
- Some current goals can no longer be considered because they are either satisfied through expected rewards, or dropped because they are hindered by the new goals that result from norm compliance.

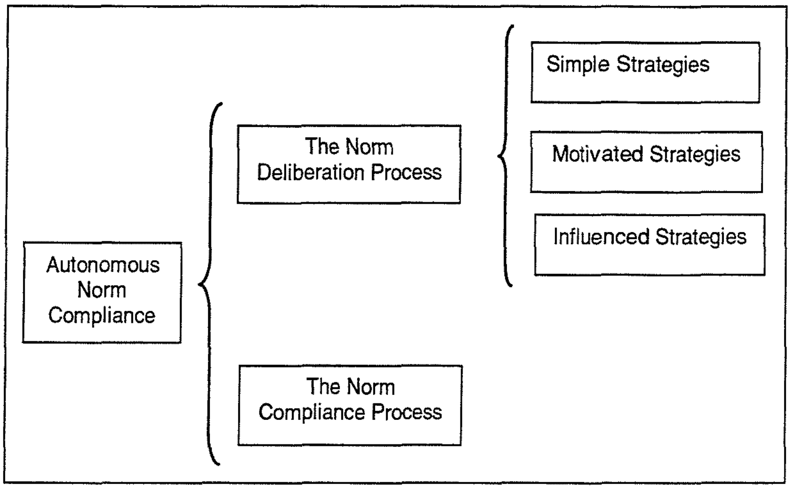

Autonomous Norm Compliance
The Norm Deliberation Process
Simple Strategies
Motivated Strategies
Influenced Strategies
The Norm Compliance Process

FIGURE 7.1: Norm Compliance Components

The different aspects of autonomous norm compliance are illustrated in Figure 7.1, and their details are explained in different sections of this chapter. First, the processes that comprise norm deliberation are described in Section 7.2. It includes discussions about whether punishments and rewards might affect the goals of an agent, and specifies the operations to accept or reject compliance with a norm. Simple strategies to decide whether to comply with a norm are discussed in Section 7.3, whereas Section 7.4 discusses motivated strategies. Further strategies based on the observed behaviour of other agents (influenced strategies) are explained in Section 7.5. Section 7.6 explains the process for complying with a norm, i.e. it describes how the goals of an agent are updated according to an agent's normative decisions. Section 7.8 describes the modelling of typical normative agents by combining different strategies of norm compliance. Finally, the contributions and conclusions of this chapter are presented.

7.2 Norm Deliberation

7.2.1 Dealing with Social Pressure

As was discussed earlier, *enforcement mechanisms* are needed as a means of ensuring that personal interests do not overcome social rules. Usually, enforcement mechanisms are associated with punishments and rewards so that agents are motivated to obey norms because of either the fear of being punished or the desire for acquiring something without apparent effort. However, as some sociologists point out [104], punishments and rewards will only affect an agent's decision to comply with norms if they either hinder or benefit one of the agent's goals. That is, punishments are not useful if none of the agent's interests (translated as individual goals) is hindered. For example, the norm of wearing fashionable clothes may have an associated punishment of not being socially accepted. However, this applies just to a specific group of agents, and there may be others less interested in social acceptance who therefore consider the fulfillment of that norm as unworthy. Similarly, rewards are a means to motivate agents only if one of an agent's goals receives benefit from them. For example, when a mother promises her child to give him a spinach pie if he cleans his room, the pie does not have any effect if the child doesn't like spinach.

We state that when deliberating about a norm, agents must consider the overarching effects of their punishments and rewards on their individual goals. In the model of norms proposed in this thesis both punishments and rewards are defined as sets of goals. Thus, agents must assess the set of their goals that can be *hindered* by one of the goals of the set of punishments, and the set of goals that can *benefit* from any of the goals of the set of rewards associated with a norm.

7.2.2 Normative Agent State

At any particular time, *active* norms are the only set of norms considered by agents to be fulfilled. These are all the norms that agents believe must be complied with in the current state. An active norm is *non-conflicting* if its compliance does not cause any conflict with one of the agent's current goals. Thus, no goals of the addressee agent are hindered by satisfying the normative goals of the norm. By contrast, an active norm is *conflicting* if its fulfillment hinders any of the agent's goals. The set of active norms is divided into two disjoint sets of norms, and agents can use a different strategy on each set of norms. For instance, agents might decide to comply with all non-conflicting norms and reject the conflicting ones. The division of norms before deliberation is illustrated, by using a Venn diagram, in Figure 7.2.

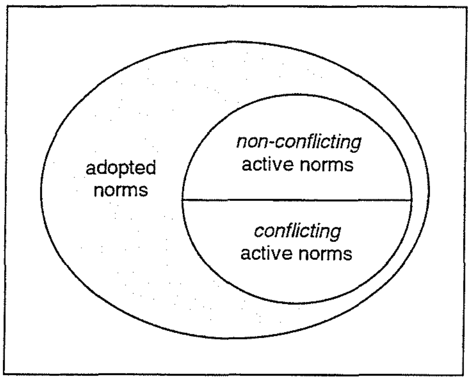


FIGURE 7.2: Division of Norms before Deliberation

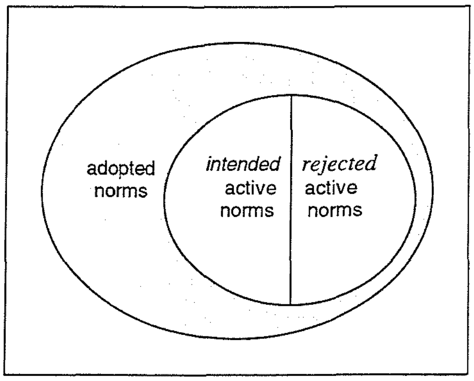


FIGURE 7.3: Division of Norms after Deliberation

By contrast, after norm deliberation, the set of active norms will be divided into two other disjoint sets of norms: *intended* norms (which represent those active norms that the agent has decided to comply with) and *rejected* norms. The Venn diagram in Figure 7.3 illustrates this different division of active norms.

Summarising, active norms are divided into conflicting and non-conflicting norms so that different or similar strategies for norm-compliance are applied to them. During deliberation, some of the conflicting norms are accepted to be intended and others are rejected. Something similar occurs for non-conflicting norms. After norm deliberation, the set of intended norms consists of those conflicting and non-conflicting norms that are accepted to be complied with by the agent, and the set of rejected norms consists of all conflicting and non-conflicting norms that are rejected by the agent. The elements in the sets of intended and rejected norms depend on the selected strategy for norm compliance. The different strategies are explained later on. Now, an agent able to decide which norms to intend and which norms to reject must be defined.

The state of an agent that has selected the norms it is keen to fulfill is formally represented in the *NormativeAgentState* schema. It represents a normative agent with no conflicting goals and norms, together with the variables representing the sets of *active*, *intended*, and *rejected* norms at a particular point of time. The predicate *conflicting* holds for a norm if and only if its normative goals conflict with any of the agent's current goals. This will be useful later on when specifying the different strategies for norm compliance. The first predicate in the schema states that active norms are a subset of all norms already adopted by the agent, whereas the second indicates that active norms are all the norms the agent believes must be complied with in the current state (i.e. those norms for which the context matches the beliefs of the agent). The third predicate states that the set of active norms has already been assessed and divided into norms to intend and norms to reject. The state of an agent is consistent in that its current goals do not conflict with the intended norms and, consequently, no normative goal must be in conflict with current goals (stated in the fourth predicate). Moreover, since rewards benefit the achievement of some goals, which means that agents do not have to work on their satisfaction because someone else does, these goals must not be part of the goals of an agent (in the fifth predicate). The sixth predicate states that punishments must be accepted and, consequently, none of the goals of an agent must hinder them. The final predicate is the definition of *conflicting* norms, which are norms whose normative goals hinder some of the agent's goals. Both *hindered* and *benefited* predicates are defined in Subsection 5.4.2.

$$
\begin{array}{|l}
\hline
\mathit{NormativeAgentState} \\
\hline
\mathit{NormativeAgent} \\
\mathit{activenorms} : \mathbb{P}\,\mathit{NormInstance} \\
\mathit{intendednorms} : \mathbb{P}\,\mathit{NormInstance} \\
\mathit{rejectednorms} : \mathbb{P}\,\mathit{NormInstance} \\
\mathit{conflicting}\,\_ : \mathbb{P}\,\mathit{NormInstance} \\
\hline
\mathit{activenorms} \subseteq \mathit{norms} \\
\forall\, \mathit{an} : \mathit{activenorms} \bullet \mathit{activenorm}\ (\mathit{an}, \mathit{beliefs}) \\
\mathit{activenorms} = \mathit{intendednorms} \cup \mathit{rejectednorms} \\
\mathit{hindered}(\mathit{goals}, \mathit{normgoals}\ \mathit{intendednorms}) = \varnothing \\
\mathit{benefited}(\mathit{goals}, \mathit{rewardgoals}\ \mathit{intendednorms}) \cap \mathit{goals} = \varnothing \\
\mathit{hindered}(\mathit{goals}, \mathit{punishgoals}\ \mathit{rejectednorms}) = \varnothing \\
\forall\, n : \mathit{activenorms} \bullet \mathit{conflicting}\ n \Leftrightarrow \\
\qquad \mathit{hindered}(\mathit{goals}, n.\mathit{normativegoals}) \neq \varnothing \\
\hline
\end{array}
$$

7.2.3 Norm Acceptance Processes

Though the particular strategies to select a norm to be intended or rejected are explained later on, a general process to comply with a norm, and a general process to reject a norm can be defined now. Thus, for a norm to be intended some constraints must be fulfilled, as follows. First, the agent must be an addressee of the norm. Then, the norm must be an adopted and currently active norm, and it must not be already intended. In addition, the agent must believe that it is not in an *exception* state and, therefore, it must comply with the norm. Formally, the process to accept a single norm as input (*newnorm*?) to be complied with is specified in the *NormIntend* schema. The first five predicates represent the constraints on the agent and the norm as described above. The sixth predicate represents the addition of the accepted norm to the set of intended norms while the set of rejected norms remains the same (final predicate).

NormIntend

$newnorm? : NormInstance$

$\Delta NormativeAgentState$

$self \in newnorm?.addressees$

$newnorm? \in norms$

$newnorm? \in activenorms$

$newnorm? \notin intendednorms$

$\neg\ logicalconsequence(beliefs, newnorm?.exceptions)$

$intendednorms' = intendednorms \cup \{newnorm?\}$

$rejectednorms' = rejectednorms$

The process for rejecting an active norm is similarly defined. To consider a norm to be rejected, the agent must be an addressee of it, the norm must be an adopted and active norm, it must not already be intended, and the agent must not be in an exception state. This is formally represented in the *NormReject* schema where the constraints on both the norm and the agent are represented in the first four predicates. The last two predicates state that, in this case, the set of intended norms remains the same and *newnorm*? norm is added to the set of reject norms.

NormReject

$newnorm? : Norm$

$\Delta NormativeAgentState$

$self \in newnorm?.addressees$

$newnorm? \in norms$

$newnorm? \in activenorms$

$newnorm? \notin intendednorms$

$\neg\ logicalconsequence(beliefs, newnorm?.exceptions)$

$intendednorms' = intendednorms$

$rejectednorms' = rejectednorms \cup \{newnorm?\}$

A norm can also be rejected because the agent is in an exception state. To distinguish this case from the case in which agents reject norms by their own decision, we introduce the *NormRejectException* schema below. The difference between both schemas is the exception state in which agents believe they are.

$$
\begin{array}{|l}
\textit{NormRejectException} \\
\hline
\mathit{newnorm?} : \mathit{Norm} \\
\Delta \mathit{NormativeAgentState} \\
\hline
\mathit{self} \in \mathit{newnorm?}.\mathit{addressees} \\
\mathit{newnorm?} \in \mathit{norms} \\
\mathit{newnorm?} \in \mathit{activenorms} \\
\mathit{newnorm?} \notin \mathit{intendednorms} \\
\mathit{logicalconsequence}(\mathit{beliefs}, \mathit{newnorm?}.\mathit{exceptions}) \\
\mathit{intendednorms}' = \mathit{intendednorms} \\
\mathit{rejectednorms}' = \mathit{rejectednorms} \cup \{\mathit{newnorm?}\} \\
\hline
\end{array}
$$

Note that in all these processes the value of *newnorm?* is generated internally when considering a norm, though it is specified as an external input to the operations for now. The processes represented in the *NormIntend* and *NormReject* schemas are used in combination with different strategies (explained in subsequent sections) to describe how agents decide whether a norm should be fulfilled. In these strategies, a new norm is evaluated depending on if it is a conflicting or non-conflicting norm.

7.3 Simple Strategies

There are different ways for agents to decide whether a norm must be rejected or complied with, we call them *strategies for norm-compliance*. These strategies differ from each other in the kinds of elements that agents consider to take a decision. We have classified them in three groups: simple, motivated and influenced strategies. In this section, simple strategies are explained.

Simple strategies are those in which norms are either intended or rejected by considering few (or no) elements of a norm. Agents that use these strategies do not make any deliberation about the effects that compliance with a norm might have on their goals. This group comprises social, fearful, greedy, and rebellious strategies.

7.3.1 Social

When an agent is strongly motivated by its social concerns, and its social responsibility is more important than any of its personal goals, all its norms (either conflicting or non-conflicting) are fulfilled. We say that the agent is being *social.* Social agents will never be punished and will receive the maximum social benefits provided by rewards. However, this can result in the loss of a considerable number of existing goals if the majority of the normative goals conflict with them. Formally, we use the *NormIntend* schema to accept a norm, together with a predicate to state that the norm can be either non-conflicting or conflicting, even though it is accepted.

$$
\begin{array}{|l}
\multicolumn{1}{l}{\mathit{SocialNCComply}} \\ \hline
\mathit{NormIntend} \\ \hline
\neg\ \mathit{conflicting}\ \mathit{newnorm?} \\ \hline
\end{array}
$$

$$
\begin{array}{|l}
\multicolumn{1}{l}{\mathit{SocialCComply}} \\ \hline
\mathit{NormIntend} \\ \hline
\mathit{conflicting}\ \mathit{newnorm?} \\ \hline
\end{array}
$$

7.3.2 Fearful

The *fear* to be punished might be a motivation to fulfill norms. A simple fearful strategy means that an agent decides to comply with a norm only if such a norm includes a punishment. Fearful agents comply with norms even if punishments do not affect any of their goals. Formally, the condition to intend a norm is that the set of punishments is not an empty set, otherwise the norm is rejected.

$$
\begin{array}{|l}
\multicolumn{1}{l}{\mathit{FearfulComply}} \\ \hline
\mathit{NormIntend} \\ \hline
\mathit{newnorm?.punishments} \neq \varnothing \\ \hline
\end{array}
$$

$$
\begin{array}{|l}
\multicolumn{1}{l}{\mathit{FearfulReject}} \\ \hline
\mathit{NormReject} \\ \hline
\mathit{newnorm?.punishments} = \varnothing \\ \hline
\end{array}
$$

7.3.3 Greedy

Greed might be another motivation to comply with a norm. In a somewhat symmetric manner to fearful agents, the *greedy strategy* means that agents obey norms only if they receive something in exchange, even though none of their goals benefit from the associated rewards. Formally, a norm is intended if the set of rewards is not empty, otherwise the norm is rejected

GreedyComply

NormIntend

$newnorm?.rewards \neq \varnothing$

GreedyReject

NormReject

$newnorm?.rewards = \varnothing$

7.3.4 Rebellious

Finally, the situation in which agents reject any social norms is described. In *rebellious* behaviour, agents may refuse to obey external orders even though by neglecting norms some of their goals could be hindered by the applied punishments. This is a kind of anti-social behaviour, and rebellious agents simply reject all norms. Formally, this is represented by using the schema to reject norms, and constraints to specify that the norm can be either non-conflicting or conflicting.

NCRebellious

NormReject

$\neg\ conflicting\ newnorm?$

CRebellious

NormReject

$conflicting\ newnorm?$

7.4 Motivated Strategies

Agents that use motivated strategies consider the possible effects on their goals of both the normative goals and the rewards in the case a norm is fulfilled, and the effects of punishments if it is not. To comply with the norm, agents assess two things: the goals that might be hindered by satisfying the normative goals, and the goals that might benefit from the associated rewards. By contrast, to reject a norm, agents evaluate the damaging effects of punishments (i.e. the goals hindered due to the satisfaction of the goals associated with punishments.) Since the satisfaction of some of their goals might be prevented in both cases, agents use the *importance* of their goals (as defined in Subsection 3.5.2) to make a decision.

As mentioned early in this thesis, the importance of a goal is determined by its associated motivations. Thus, the higher the intensity of the motivations the more important the goal. At run time, goal importance is used to decide which goal should be achieved first, i.e. the most motivated of the goals must be intended first. When deliberating about norms, goal importance is used for deciding which goal an agent prefers to hold because in complying with its duties some some personal goals might not be satisfied.

We state that, norms are violated when their fulfillment hinders personal goals that agents consider as worthy for their personal interest. For example, an obligation of paying taxes may frustrate the personal goals of taking holidays abroad. In this case, the decision concerns only the agent which, based on its motivations and current situation, must decide what is more important. Some careless agents may take this decision just by considering both the normative goals and their personal goals, but others may also take into consideration the consequences of being either punished or rewarded. For example, if an agent decides not to pay its taxes and continues with its goal towards some enjoyable holidays, it must accept the consequences of being fined, and therefore spending much more money in paying both fines and taxes. Cautious agents consider both the possibility of being punished and how much these punishments may affect their other personal goals. In the remainder of this section, two strategies in which an agent's motivations play an important role are described in detail.

7.4.1 Pressured

Sometimes the fulfillment of a norm is considered as a last resort in order to avoid some personal goals being prevented by sanctions. Agents are *pressured* to obey norms through the application of punishments that might hinder some of their important goals. Agents that adopt a pressured strategy face four different cases.

1. The norm is a non-conflicting norm and some goals are hindered by its punishments.
2. The norm is a non-conflicting norm and there are no goals hindered by its punishments.
3. The norm is a conflicting norm and the goals hindered by its normative goals are less important than the goals hindered by its punishments.
4. The norm is a conflicting norm and the goals hindered by its normative goals are more important than the goals hindered by its punishments

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

$$
\begin{array}{|l}
\hline
\textit{PressuredNCComply} \\
\textit{NormIntend} \\
\hline
\neg\ \textit{conflicting newnorm?} \\
\textit{hindered}(\textit{goals}, \textit{newnorm?.punishments}) \neq \varnothing \\
\hline
\end{array}
$$

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

$$
\begin{array}{|l}
\hline
\textit{PressuredNCReject} \\
\textit{NormReject} \\
\hline
\neg\ \textit{conflicting newnorm?} \\
\textit{hindered}(\textit{goals}, \textit{newnorm?.punishments}) = \varnothing \\
\hline
\end{array}
$$

According to our definition, a conflicting norm is a norm whose normative goals hinder an agent's goals. In this situation, agents comply with the norm at the expense of existing goals only if what they can lose through punishments is more important than what they can lose by complying with the norm. Formally, a conflicting norm is intended if the goals that could be hindered by punishments (*hps*) are more important than the set of existing goals hindered by normative goals (*hngs*). This is represented in the *PressuredCComply* schema where the *importance* function uses the motivations associated with the set of goals to find the importance of goals as described in Subsection 3.5.2.

$$\begin{array}{|l}
\textit{PressuredCComply} \\
\hline
\textit{NormIntend} \\
\hline
\textit{conflicting newnorm?} \\
\textbf{let } \textit{hps} == \textit{hindered}(\textit{goals}, \textit{newnorm?.punishments}) \bullet \\
\textbf{let } \textit{hngs} == \textit{hindered}(\textit{goals}, \textit{newnorm?.normativegoals}) \bullet \\
\qquad \textit{importance}\ (\textit{gms}, \textit{hps}) > \textit{importance}\ (\textit{gms}, \textit{hngs}) \\
\hline
\end{array}$$

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

$$\begin{array}{|l}
\textit{PressuredCReject} \\
\hline
\textit{NormReject} \\
\hline
\textit{conflicting newnorm?} \\
\textbf{let } \textit{hps} == \textit{hindered}(\textit{goals}, \textit{newnorm?.punishments}) \bullet \\
\textbf{let } \textit{hngs} == \textit{hindered}(\textit{goals}, \textit{newnorm?.normativegoals}) \bullet \\
\qquad \textit{importance}\ (\textit{gms}, \textit{hps}) \leq \textit{importance}\ (\textit{gms}, \textit{hngs}) \\
\hline
\end{array}$$

These four cases are summarised in Figure 7.4, where *hps* and *hngs* represent the sets of goals that can be hindered by punishments or normative goals, respectively.

7.4.2 Opportunistic

There are also *opportune* situations where the fulfillment of a norm may contribute to the achievement of some of the agent's goals. That is, compliance with norms is ensured through the benefits obtained by addressees from the rewards. Agents that use an opportunistic strategy consider four cases to take a decision accordingly.

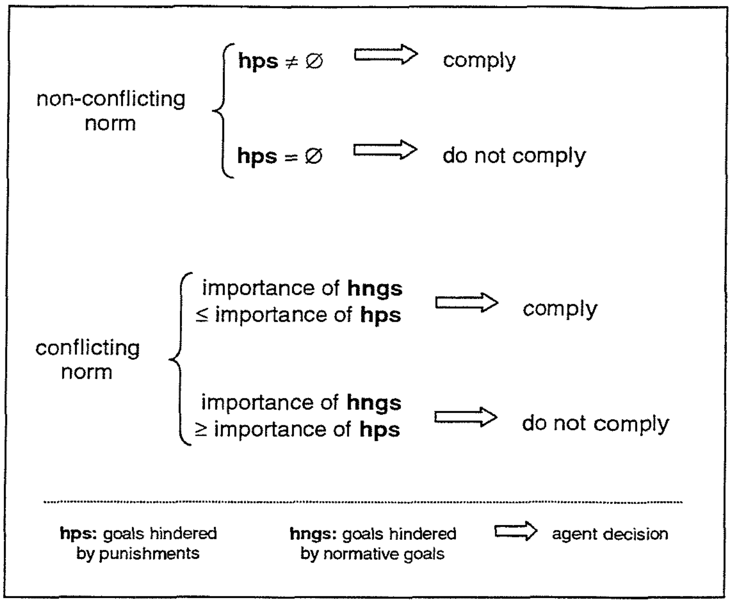

non-conflicting norm
hps ≠ ∅
comply
hps = ∅
do not comply
conflicting norm
importance of hngs ≤ importance of hps
comply
importance of hngs ≥ importance of hps
do not comply
hps: goals hindered by punishments
hngs: goals hindered by normative goals
agent decision

FIGURE 7.4: Pressured Norm Compliance

1. The norm is a non-conflicting norm and some goals can benefit from rewards.
2. The norm is a non-conflicting norm and there are no goals that benefit from rewards.
3. The norm is a conflicting norm and the goals hindered by its normative goals are less important than the goals that benefit from rewards.
4. The norm is a conflicting norm and the goals hindered by its normative goals are more important than the goals that benefit from rewards.

The first case represents those norms whose fulfillment might provide benefits for an agent's goals rather than cause damage to them. No goals are hindered by satisfying the normative goals, because the norm is non-conflicting, but some goals can benefit from rewards. Consequently, the best decision is to intend the norm. Formally, non-conflicting norms are fulfilled only if their rewards benefit some goals. Again, the *NormIntend* operation schema to accept a norm is used by adding the corresponding constraints on norms as follows.

OpportunisticNCComply

NormIntend

$\neg\ \mathit{conflicting\ newnorm?}$

$\mathit{benefited}(\mathit{goals}, \mathit{newnorm?.rewards}) \neq \varnothing$

In the second case, there are no goals hindered by normative goals and none of an agent's goals benefits from rewards. Consequently, the norm is rejected. To formalise this, the *NormReject* operation schema is used to reject a non-conflicting norm whose rewards do not benefit some goals.

$$\begin{array}{|l}
\multicolumn{1}{l}{\_\,\mathit{OpportunisticNCReject}\,\rule{12em}{0.4pt}} \\
\mathit{NormReject} \\
\hline
\neg\ \mathit{conflicting}\ \mathit{newnorm?} \\
\mathit{benefited}(\mathit{goals}, \mathit{newnorm?.rewards}) = \varnothing \\
\hline
\end{array}$$

When faced with conflicting norms, an agent's motivations determine how it acts depending on the kind of rewards offered. In the two cases of conflicting norms, if agents comply, they might lose some goals, but other goals might benefit from rewards. In the third case, if agents comply with the norm they gain more than they lose otherwise and, therefore, the norm is accepted. Formally, conflicting norms are complied with only when their associated rewards benefit goals (*brs*) that are more important than those hindered by the normative goals(*hngs*). This is formalised as follows.

$$\begin{array}{|l}
\multicolumn{1}{l}{\_\,\mathit{OpportunisticCComply}\,\rule{12em}{0.4pt}} \\
\mathit{NormIntend} \\
\hline
\mathit{conflicting}\ \mathit{newnorm?} \\
\mathbf{let}\ \mathit{brs} == \mathit{benefited}(\mathit{goals}, \mathit{newnorm?.rewards}) \bullet \\
\mathbf{let}\ \mathit{hngs} == \mathit{hindered}(\mathit{goals}, \mathit{newnorm?.normativegoals}) \bullet \\
\qquad \mathit{importance}\ (\mathit{gms}, \mathit{brs}) > \mathit{importance}\ (\mathit{gms}, \mathit{hngs}) \\
\hline
\end{array}$$

By contrast, if it is more important to preserve those goals that could be hindered by normative goals than to receive the benefits of rewards, the norm is rejected. Formally, a conflicting norm is rejected if the goals that benefit from rewards (*brs*) are less important than the goals hindered by the normative goals (*hngs*). Figure 7.5 summarises the four cases comprising the opportunistic strategy.

OpportunisticCReject

NormReject

---

conflicting newnorm?

let $brs == benefited(goals, newnorm?.rewards) \bullet$

let $hngs == hindered(goals, newnorm?.normativegoals) \bullet$

$\quad importance\ (gms, brs) \leq importance\ (gms, hngs)$

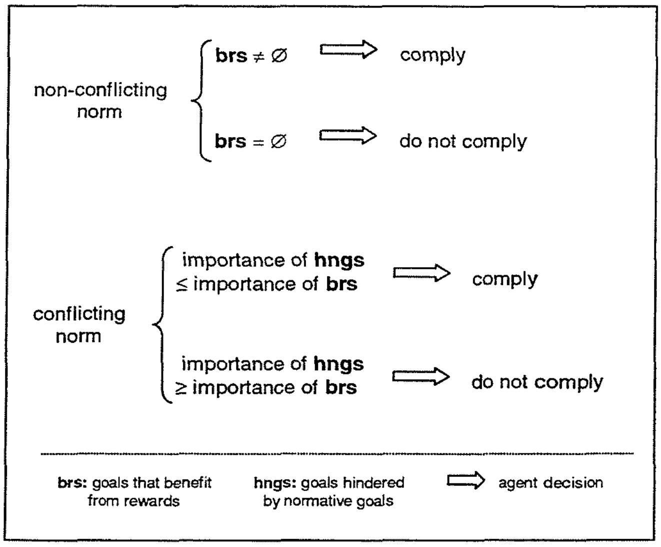

non-conflicting norm
brs ≠ ∅
comply
brs = ∅
do not comply
conflicting norm
importance of hngs ≤ importance of brs
comply
importance of hngs ≥ importance of brs
do not comply
brs: goals that benefit from rewards
hngs: goals hindered by normative goals
agent decision

FIGURE 7.5: Opportunistic Norm Compliance

Combinations of these strategies are also possible. For example, an agent can be *pressured* and *opportunistic* and therefore *selfish* because it only fulfills a norm when one of its interests is either threatened by punishments or benefits from rewards. Details of how this can be done are given later on in Section 7.8.

7.5 Influenced Strategies

Agents take decisions not only based on their own goals and motivations, but also by observing the normative behaviour of other agents. Influenced strategies for norm compliance are those in which agents are influenced, to comply with a norm, by the norm-related actions of other agents. For example, agents could decide either to comply with a norm that everyone in the society has already complied with, or to reject a norm that nobody complies with. In this section, three different influenced strategies are explained.

Before describing how the decision to comply with norms might be influenced by the normative behaviour of other agents, agents that have selected the norms they are keen to fulfill, and that are able to observe the society in which they participate, and to keep records of past compliance with norms, must be modelled. To do so, we introduce the *SocietyAgent* schema which includes both the *NormativeAgentState* schema (defined earlier in this chapter) and the *SocietyNormativeAgent* schema (defined in Subsection 5.4.3).

SocietyAgent

NormativeAgentState
SocietyNormativeAgent

7.5.1 Simple Imitation

Agents follow a *simple imitation* strategy as part of their desire to be like other agents. In this case, if other addressee agents comply with a norm, these agents also comply with it. Similarly, if other addressee agents violate a norm, these agents violate it. The strategy can be used by agents, for example, when they are in unknown environments, and no other criteria for decision-making are available. Formally, an agent complies with a norm if there exists an instance of the same norm which is already fulfilled (that means an addressee has already complied with it). This is represented in the schema below, where both the *SocietyAgent* schema (to include the states of both the agent and the norms in its society) and the *NormIntend* schema (to accept a norm) are included. The predicate states that the norm addressed to the agent (*newnorm*?) and the norm already fulfilled (*fn*) are instances of the same norm (*n*).

SimpleImitationComply

SocietyAgent
NormIntend

$$
\begin{array}{l}
\exists n : society.generalnorms \bullet (isnorminstance\ (newnorm?, n) \land \\
\quad (\exists fn : (societystate.fulfillednorms) \bullet isnorminstance\ (fn, n)))
\end{array}
$$

Agents that follow a simple imitation strategy reject a norm if there exists someone who has already rejected it. Formally, an agent rejects a norm if there exists an instance of the same norm which is unfulfilled (which means an addressee of the norms has not complied with it). This is represented in the schema below where the predicate states that both the rejected norm (*newnorm*?) and the unfulfilled norm (*un*) are instances of the same norm (*n*).

$$
\begin{array}{|l}
\hline
\multicolumn{1}{l}{\textit{SimpleImitationReject}} \\
\textit{SocietyAgent} \\
\textit{NormReject} \\
\hline
\exists\, n : \textit{society.generalnorms} \bullet (\textit{isnorminstance}\ (\textit{newnorm?}, n) \wedge \\
\qquad (\exists\, \textit{un} : (\textit{societystate.unfulfillednorms}) \bullet \textit{isnorminstance}\ (\textit{un}, n))) \\
\hline
\end{array}
$$

Then, the normative behaviour of agents that imitate the normative behaviour of other addressees of the same norm is represented by the disjunction of two schemas as follows.

$$\textit{SimpleImitation} \mathrel{\widehat{=}} \textit{SimpleImitationComply} \vee \textit{SimpleImitationReject}$$

Notice that this strategy includes cases in which the agent itself has previously fulfilled the norm. The simple imitation strategy can be refined further to represent more complex cases of imitation. For example, agents might comply with those norms already complied with by the *majority* of addressee agents.

7.5.2 Reasoned Imitation

In a reasoned imitation strategy, agents not only observe the behaviour of other addressees of the norm but they also observe the behaviour of its promoters (the agents responsible for rewarding) and defenders (the agents responsible for punishing). To describe this strategy, we can consider four cases relating to different observed behaviour.

1. An addressee complied with the norm and, by doing so, was rewarded.
2. An addressee complied with the norm, and was never rewarded.
3. An addressee did not comply with the norm and, as a result, was punished.
4. An addressee did not comply with the norm, and was not punished.

In the first case, agents imitate those agents which, by fulfilling the norm, received the corresponding rewards. That is, the norm was fulfilled by one of the addressee agents and the promoter of this norm also complied with the corresponding reward norm (i.e. the rewards were given). Here, two interlocking norms were activated and fulfilled by their respective addressee agents. Formally, an agent complies with a norm if there exists an instance of the same norm that is both fulfilled and rewarded. This is represented in the *RewardedImitationComply* schema, where the predicate states that the norm (*newnorm?*) addressed to the agent and the norm already fulfilled (*fn*) and rewarded are instances of the same norm (*n*).

RewardedImitationComply

SocietyAgent
NormIntend

$$\begin{array}{l}\exists\, n : \mathit{society.generalnorms} \bullet (\mathit{isnorminstance}\ (\mathit{newnorm}?, n) \land \\ \quad (\exists\, \mathit{fn} : (\mathit{societystate.fulfillednorms}) \bullet \\ \qquad (\mathit{fn} \in \mathit{societystate.rewardednorms}) \land \mathit{isnorminstance}\ (\mathit{fn}, n)))\end{array}$$

By contrast, the second case describes situations in which, despite an agent complying with the norm, the deserved rewards were never given. Agents that observe this case prefer to dismiss such a norm. Formally, an agent rejects a norm if there exists an instance of the same norm that is both fulfilled and not rewarded. This is represented in the *RewardedImitationReject* schema, where the predicate states that the norm (*newnorm?*) addressed to the agent and the norm already fulfilled (*fn*) and not rewarded are instances of the same norm (*n*).

RewardedImitationReject

SocietyAgent
NormReject

$$\begin{array}{l}\exists\, n : \mathit{society.generalnorms} \bullet (\mathit{isnorminstance}\ (\mathit{newnorm}?, n) \land \\ \quad (\exists\, \mathit{fn} : (\mathit{societystate.fulfillednorms}) \bullet \\ \qquad (\mathit{fn} \notin \mathit{societystate.rewardednorms}) \land \mathit{isnorminstance}\ (\mathit{fn}, n)))\end{array}$$

In the third case, agents observe the bad consequences of dismissing a norm by an offender. Here, an addressee of the same norm did not comply with the norm and, as a result, was punished by a defender of the norm. Agents that observe this event prefer to comply with the norm and to avoid something similar occuring to them. Formally, an agent complies with a norm if there exists an instance of the same norm that is both unfulfilled and punished. The *PunishedImitationComply* schema represents this situation. Its predicate states that the norm (*newnorm?*) addressed to the agent and the unfulfilled norm (*un*), which is also a punished norm, are instances of the same norm (*n*).

PunishedImitationComply

SocietyAgent
NormIntend

$$
\begin{array}{l}
\exists\, n : \mathit{society.generalnorms} \bullet (\mathit{isnorminstance}\ (\mathit{newnorm?}, n) \land \\
\quad (\exists\, \mathit{un} : (\mathit{societystate.unfulfillednorms}) \bullet \\
\qquad (\mathit{un} \in \mathit{societystate.punishednorms}) \land \mathit{isnorminstance}\ (\mathit{un}, n)))
\end{array}
$$

By contrast, if offenders of a norm are never punished, they might be a bad influence for other addressees of the same norm that might decide to do the same. This is the last case considered by this strategy. Formally, an agent rejects a norm if there exists an instance of the same norm that is both unfulfilled and not punished. The representation of this is given in the *PunishedImitationReject* schema, where the predicate states that the norm (*newnorm*?) addressed to the agent and the unfulfilled norm (*un*), which is also a punished norm, are instances of the same norm (*n*).

PunishedImitationReject

SocietyAgent
NormReject

$$
\begin{array}{l}
\exists\, n : \mathit{society.generalnorms} \bullet (\mathit{isnorminstance}\ (\mathit{newnorm?}, n) \land \\
\quad (\exists\, \mathit{un} : (\mathit{societystate.unfulfillednorms}) \bullet \\
\qquad (\mathit{un} \notin \mathit{societystate.punishednorms}) \land \mathit{isnorminstance}\ (\mathit{un}, n)))
\end{array}
$$

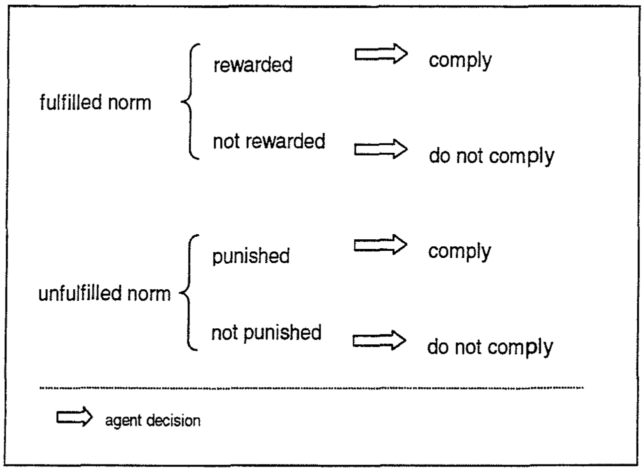


FIGURE 7.6: Reasoned Imitation Norm Compliance

Notice that these strategies only work for those norms that include explicit punishments and rewards. The cases are summarised in Figure 7.6. Finally, an agent that observes the normative behaviour of other addressees, defenders and promoters of the norm to decide whether to comply with the norm is represented in the *ReasonedImitation* schema. Here, agents either comply with or reject a norm as explained above.

$$
\begin{aligned}
&\mathit{ReasonedImitation} \mathrel{\widehat{=}} \\
&\qquad (\mathit{RewardedImitationComply} \land \mathit{PunishedImitationComply}) \lor \\
&\qquad (\mathit{RewardedImitationReject} \land \mathit{PunishedImitationReject})
\end{aligned}
$$

7.5.3 Reciprocation

From the sociological perspective, *reciprocity* is a principle that makes human relationships stable and maintains the cohesion and equilibrium in a society [82]. Reciprocity states that agents have to provide help to those agents that have helped them. This principle has been already introduced by others as a means to promote cooperation among self-interested computational agents [149, 150]. Here, we introduce it as a strategy to decide whether to comply with a norm, and we call it the *reciprocation* strategy. By using this strategy, an agent complies with a norm if by doing so, agents from whom it has received some benefits in the past, now benefit from its actions. The relationship between these two agents is illustrated in Figure 7.7.

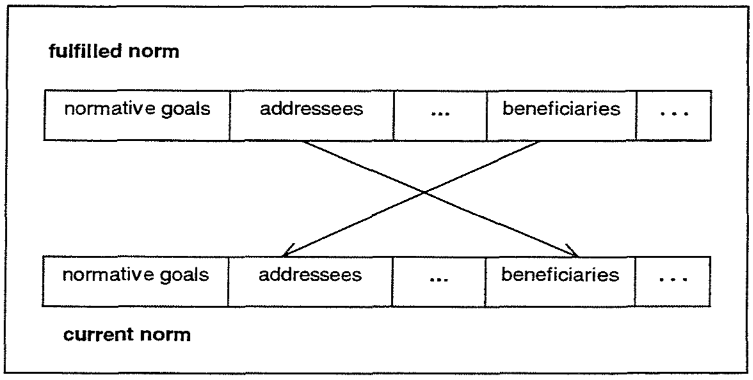


FIGURE 7.7: Agents in Reciprocation Relationship

Formally, an agent accepts a norm if its corresponding beneficiaries are addressees of a fulfilled norm for which the agent is a beneficiary. The specification is given in the *ReciprocationComply* schema, where the *SocietyAgent* and *NormIntend* schemas are included as usual. The predicate states that beneficiaries (*bag*) of the accepted norm (*newnorm*?) must be addressees of a fulfilled norm (*fn*) for which the agent is a beneficiary.

$$
\begin{array}{|l}
\textit{ReciprocationComply} \\
\hline
\textit{SocietyAgent} \\
\textit{NormIntend} \\
\hline
\forall\, bag : newnorm?.beneficiaries \bullet (\exists\, fn : societystate.fulfillednorms \bullet \\
\qquad (bag \in fn.addressees \land self \in fn.beneficiaries)) \\
\hline
\end{array}
$$

Similarly, agents that follow this strategy reject any norm whose benefits can be enjoyed by agents that did not fulfill norms that could benefit them. Formally, an agent rejects a norm if its corresponding beneficiaries are addressees of an unfulfilled norm for which the agent is a beneficiary. This is specified in the *ReciprocationReject* schema, where the predicate states that beneficiaries (*bag*) of the rejected norm (*newnorm*?) must be addressees of an unfulfilled norm (*un*) for which the agent is a beneficiary.

$$
\begin{array}{|l}
\textit{ReciprocationReject} \\
\hline
\textit{SocietyAgent} \\
\textit{NormReject} \\
\hline
\exists\, bag : newnorm?.beneficiaries \bullet (\exists\, un : societystate.unfulfillednorms \bullet \\
\qquad (bag \in un.addressees \land self \in un.beneficiaries)) \\
\hline
\end{array}
$$

By contrast with simple imitation strategies, in the reciprocation strategy the observed behaviour does not correspond to addressees of the same norm, but to addressee agents of norms that benefit the agent. This strategy might be more suitable to apply to those norms without an assigned defender, where the agent itself rewards or punishes past compliance with norms. However, it might be unsuitable for those norms addressed to a large group of agents such as the laws that a government issues, or norms that agents fulfill in a structured organisation where beneficiary agents are not very well specified. In these cases, agents are convinced to comply with a norm through very well established mechanisms to punish rather than being motivated by a reciprocity principle. The complete strategy whose cases were explained above can be formalised as follows.

$$\textit{NormReciprocation} \mathrel{\widehat{=}} \textit{ReciprocationComply} \lor \textit{ReciprocationReject}$$

7.6 Norm Compliance

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

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

To make the model simple, we assume that punishments are always applied, and rewards are always given though the possibility exists that agents never become either punished or rewarded. In addition, note that the set of goals hindered by normative goals can be empty if the norm being considered is a non-conflicting norm, and goals hindered by punishments or goals that benefit from rewards can be empty if a norm does not include any of them.

The process to comply with the norms an agent has decided to fulfill is specified in the *NormComply* schema. Through this process the set of goals is updated, so that all sets of normative goals (*ngs*) that correspond to the intended norms are added to them. As a result of this, any existing goals that conflict with these normative goals (*hngs*) are hindered, and must be removed. Goals that benefit from the rewards (*brs*) associated with the intended norm are also removed as a result of being achieved by other means. Finally, all goals hindered by the punishments (*hps*) of rejected norms must be removed.

$$\begin{array}{|l}
\textit{NormComply} \\
\hline
\Delta \textit{NormativeAgentState} \\
\hline
(\mathbf{let}\ ngs == \bigcup\{gs : \mathbb{P}\, Goal \mid (\exists\, n : intendednorms \bullet \\
\qquad gs = n.normativegoals)\} \bullet \\
(\mathbf{let}\ hngs == \bigcup\{gs : \mathbb{P}\, Goal \mid (\exists\, n : intendednorms \bullet \\
\qquad gs = hindered\ (goals, n.normativegoals))\} \bullet \\
(\mathbf{let}\ brs == \bigcup\{gs : \mathbb{P}\, Goal \mid (\exists\, n : intendednorms \bullet \\
\qquad gs = benefited(goals, n.rewards))\} \bullet \\
(\mathbf{let}\ hps == \bigcup\{gs : \mathbb{P}\, Goal \mid (\exists\, n : rejectednorms \bullet \\
\qquad gs = hindered\ (goals, n.punishments))\} \bullet \\
(\ goals' = (goals \cup ngs) \setminus (hngs \cup brs \cup hps))))) \\
\hline
\end{array}$$

As can be seen, although different strategies can be used to find the set of *intended* norms, the norm compliance process is similar in all of them.

7.7 The Control Architecture

Although we have considered the different aspects of normative reasoning, we have not yet brought them together in a control architecture. In this section, we *compose* the different parts to do just that. The norm adoption process (*AutonomousNormAdoption*) defined in Section 6.5, together with the processes for deliberating about a norm (*NormIntend* and *NormReject*) and the process for complying with a norm (*NormComply*), define the normative behaviour of agents. In Figure 7.8, a sequential invocation of these processes (represented by boxes) is indicated by continuous arrows, while dashed arrows represent the data flow, and mental attitudes are represented by circles. Thus, after norm adoption, instances of norms are created. Norm deliberation takes as input the adopted activated norms and selects those intended and those rejected. Then, intended and rejected norms are taken as input by norm compliance, which updates the goals of the agent according to the norms it has decided to comply with or to reject.

The formal representation of the normative behaviour of an agent can be given as a composition of schemas where the output of a schema on the left is taken as the input of the schema on the right. Norm deliberation has been divided into two processes: one to decide whether the norm must be intended, and the other to decide if it must be rejected as described in Section 7.2.

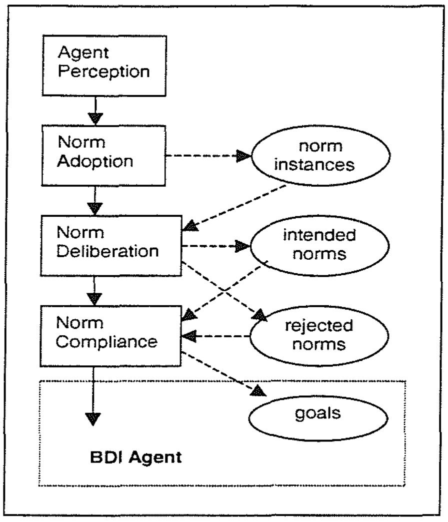


FIGURE 7.8: Agent Normative Behaviour

$$\begin{aligned}&\mathit{NormativeBehaviour} \mathrel{\widehat=} \\ &\quad \mathit{NormAdoption} \mathbin{\text{⨾}} (\mathit{NormIntend} \lor \mathit{NormReject}) \mathbin{\text{⨾}} \mathit{NormComply}\end{aligned}$$

7.8 Discussion

Figure 7.9 shows a summary of the different norm-compliance strategies. We argue that complex normative agent behaviours can be represented by applying these strategies to both non-conflicting and conflicting norms. The purpose of this section is to show how, by using these strategies together with the norm compliance process, different types of normative agents can be modelled.

Agents that always comply with norms can be modelled by specifying that the agent uses a social strategy for non-conflicting and conflicting norms as follows.

$$\mathit{SocialAgent} \mathrel{\widehat=} \mathit{SocialNCComply} \land \mathit{SocialCComply}$$

Selfish agents are agents that make decisions based on both how much they can gain and how much they can lose by complying with norms. Consequently, they can be modelled by using combinations of both pressured and opportunistic strategies as follows.

- Selfish agents intend those non-conflicting norms whose rewards benefit their goals and those non-conflicting norms whose punishments hinder their goals. Formally, this is represented as the conjunction of opportunistic and pressured strategies to comply with a non-conflicting norm as follows.

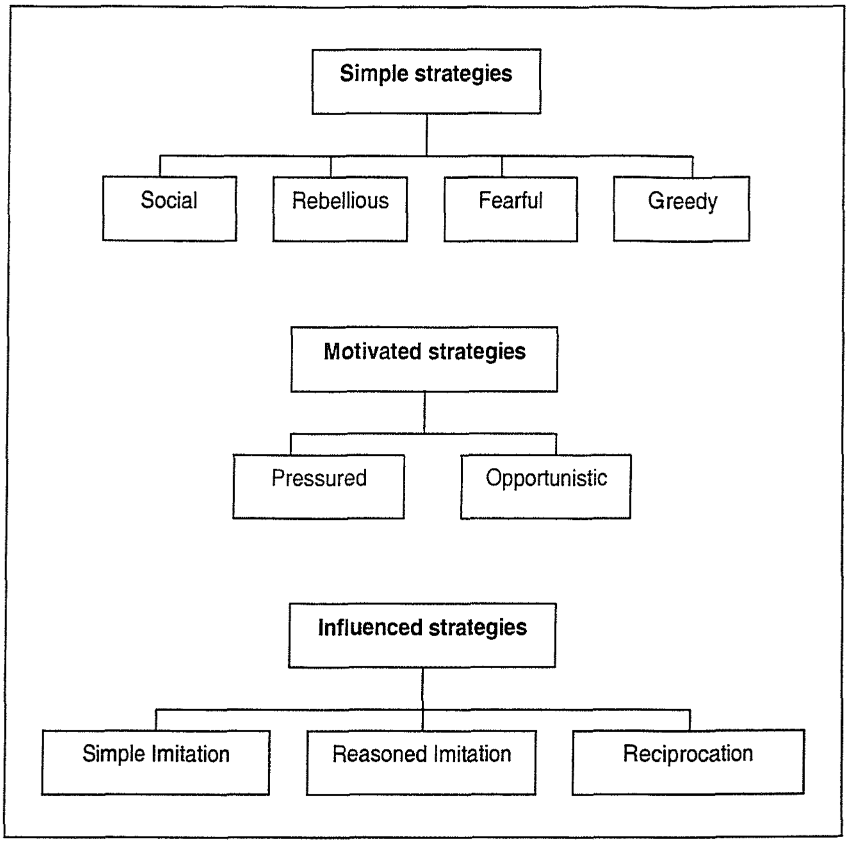


FIGURE 7.9: Strategies for Norm Compliance Decision Making

$$SelfishComplyNC \mathrel{\widehat{=}} OpportunisticNCComply \wedge PressuredNCComply$$

- Non-conflicting norms that do not affect any of their goals are rejected. This is formally defined by the conjunction of opportunistic and pressured strategies to reject a non-conflicting norm as follows.

$$SelfishRejectNC \mathrel{\widehat{=}} OpportunisticNCReject \wedge PressuredNCReject$$

- Selfish agents intend those conflicting norms whose rewards benefit more important goals than the goals hindered by normative goals. They also intend conflicting norms whose punishments hinder goals that are more important than the goals hindered by normative goals. Formally, opportunistic and pressured strategies to comply with a conflicting norm are combined as follows.

$$SelfishComplyC \mathrel{\widehat{=}} OpportunisticCComply \wedge PressuredCComply$$

- Finally, selfish agents reject those conflicting norms whose normative goals hinder their important goals. Rewards and punishments are not enough to lose an agent's goals. Formally, this is represented as the conjunction of opportunistic and pressured strategies to reject a conflicting norm.

$$\mathit{SelfishRejectC} \mathrel{\widehat{=}} \mathit{OpportunisticCReject} \land \mathit{PressuredCReject}$$

Then, the complete normative behaviour of a selfish agent is formally represented by specifying the cases in which a non-conflicting norm is either complied with or rejected, together with the cases to comply with or to reject a conflicting norm as follows.

$$\begin{aligned}\mathit{Selfish} \mathrel{\widehat{=}} {} & (\mathit{SelfishComplyNC} \lor \mathit{SelfishRejectNC}) \land \\ & (\mathit{SelfishComplyC} \lor \mathit{SelfishRejectC})\end{aligned}$$

Further combinations of strategies to decide whether to comply with a norm are also possible. For instance, to model agents that apply the reasoned imitation strategy for those conflicting norms and selfish strategies for those non-conflicting norms, we restrict the reasoned imitation strategy to be applied only to conflicting norms as follows.

$$\begin{array}{|l}
\multicolumn{1}{l}{\mathit{ReasonedComplyC}} \\
\hline
\mathit{ReasonedImitation} \\
\hline
\mathit{conflicting}\ \mathit{newnorm}? \\
\hline
\end{array}$$

After that, we can use the previously defined selfish strategy for non-conflicting norms to specify the complete normative behaviour of the required agent as follows.

$$\begin{aligned}\mathit{SelfishReasonedAgent} \mathrel{\widehat{=}} {} & (\mathit{SelfishComplyNC} \lor \mathit{SelfishRejectNC}) \land \\ & \mathit{ReasonedComplyC}\end{aligned}$$

An agent that is social for non-conflicting norms and rebellious for those that conflict with its goals can then be defined as follows.

$$\mathit{SocialRebelliousAgent} \mathrel{\widehat{=}} \mathit{SocialNCComply} \land \mathit{CRebellious}$$

Many more interesting normative behaviours can be defined by combining the different strategies proposed in this chapter. However, some combinations are not possible. For example, an agent cannot be social and rebellious for either conflicting or non-conflicting norms at the same time.

7.9 Conclusions

Agents in this model are *normative agents* with the ability to choose the set of norms they want to comply with, to satisfy these norms, and to accept the consequences of both selected and non-selected norms. Whatever decision they take about compliance, their goals may be affected not only by normative goals but also by those punishments and rewards associated with norms (the *social pressure* of norms). Compliance with norms involves an agent's *commitment* to obey them and, therefore, to satisfy their associated normative *goals*. However, since normative goals can conflict with the individual goals of agents, and they represent extra computational effort, agents must deliberate about their reasons for complying with norms.

By contrast with other proposals where agents either always comply with norms [13, 108, 153] or focus on a single strategy to identify some advantages in violating a norm [12, 54], the model for autonomous norm compliance proposed in this chapter is more general. We have clearly distinguished between the steps taken in deciding whether norms should be complied with, and the steps taken after such a decision has been made.

In our model, norm deliberation comprises a process to comply with a norm, a process to reject a norm, and examples of nine different strategies of decision. These strategies range from those in which there is no deliberation at all (simple strategies), and those based on an agent's goals and motivations (motivated strategies), to strategies in which agents are influenced by the normative behaviour of other agents. No other work covers such a range of strategies in a single framework. In addition, the consequences of normative decisions are clearly represented in a general process to comply with norms. Through this process, an agent's goals are updated accordingly with its normative decisions.

By separating the *strategies* of compliance from the general processes for accepting, rejecting and complying with a norm, our model allows an easy representation of agents that always comply with norms (social agents), agents that decide on the basis of their own interests (e.g. selfish agents), and agents that are influenced by other agents such as reciprocal agents. Furthermore, we claim that existing models of normative agents can easily be represented in our model. For instance, Conte and Castelfranchi [42] compare the behaviour of two kinds of agents, *incentive-based rational deciders* and *normative agents*, where the first comply with norms only if the utility of obedience is higher than the utility of transgression, and normative agents, by contrast, always comply. Both kinds of agents can be easily implemented in our model as *selfish* and *social* agents respectively. Similarly, Dignum et al. [54] describe a model of BDI agents in which obligations are fulfilled only if the cost of the punishment is higher than the cost of compliance. This particular view can be reduced to the case of agents that follow a *pressured* strategy. Boella and Lesmo [12] propose a strategy in which agents observe the normative behaviour of the defenders of a norm. They claim that if agents notice that the intention of defenders is to apply a punishment, agents will comply with the norm, otherwise the norm is rejected. In our model, this is not possible because, for a defender to act, is necessary that an addressee agent rejects the norm first. It does not make any sense to intend to punish an agent when defenders do not know how the agent will behave. The *reasoned imitation strategy* provides an alternative to Boella and Lesmo's strategy because there, agents observe the application of punishments to past offender of a norm as a warning signal for potential offenders.

The work developed in this chapter contributes to a better understanding of the behaviour of agents concerning compliance with norms, not only because what motivates an agent to fulfill norms has been described, but also because the way in which the fulfillment of a norm might affect an agent's goals has been explained. This is important for the design of normative agents which, although adhering to the norms of a system, can be flexible enough to react to unpredictable situations in the environment and, therefore, can reject a norm. By providing this formal approach to norm compliance some of the aspects that must be considered in order to incorporate norms into autonomous agents have been addressed. (Others aspects are considered in subsequent chapters.) Finally, we state that our model is simple and elegant, and that it facilitates the computational implementation of different types of normative agents.

Chapter 8

Norm Compliance Effects

8.1 Introduction

As defined earlier, the behaviour of normative agents is shaped by the obligations they must comply with, prohibitions that limit the kind of goals they can pursue, social commitments that are created during their social life, and social codes which may not carry punishments, but whose fulfillment is a means of being accepted by others. However, autonomous agents do not comply with norms as an end but make a decision to do so during a *norm deliberation process* in which their goals and motivations are taken into account to select the norms they want to fulfill. These agents understand that whatever decision they make, it has consequences that must be accepted as members of a society. That is, norms are supported by the society, which means that violated norms are generally punished, and fulfilled norms are, sometimes, rewarded. Furthermore, decisions regarding norms not only affect the performance of an agent, but also the performance of those agents that are expecting the norms to be fulfilled, especially when these agents are beneficiaries of the norm. Moreover, since norms are a means for social control, compliance with norms affects the effectiveness of such control.

In previous chapter, we have identified different strategies that agents use to decide whether to comply with a norm. Some intuitive ideas can be provided regarding the effects of these normative decisions on both societies and their members. For instance, we can say that societies in which individuals always fulfill norms are more stable, and that societies with rebellious individuals tends to collapse [42]. We might even think that selfish individuals are more successful regarding the satisfaction of their goals than social agents. However, a better way to explain and verify these effects is through experimental methods that simulate the normative behaviour of agents, and that allow the visualisation and comparison of results.

In analysing the effects of normative decisions, two sides of norms must be taken into account. On the one hand, since norms are issued to constrain the behaviour of agents, it is always expected that some norms conflict with the goals of agents. On the other, it is through norms that agents can satisfy some of their goals, because norms prescribe what other agents ought to do to benefit the first. Consequently, any decision regarding norms affects not only the goals of addressee agents but also the goals of beneficiaries (and probably the goals of defenders and promoters) of a norm. We argue that three different effects of autonomous norm compliance can be observed in a society whose members interact through norms.

- In complying with norms, an agent's goals become either satisfied or unsatisfied. Some of its goals may be frustrated in order to comply with its duties but, at the same time, other goals might benefit from rewards. Now, in the case an agent decides to violate a norm, punishments might be applied, and this causes some of its goals to be unsatisfied. Moreover, there are also goals whose satisfaction depends on other agents complying with norms. We say that the *individual performance* of an agent depends on the number of its goals that can be satisfied at the same time that it is being constrained by norms.

- Societies issue norms which are the means to exert social control on their members. Consequently, it is expected that all members of a society comply with norms. However, sometimes agents violate norms, and this reduces the effectiveness of the social control. We say that an agent's *contribution* to its society depends on the rate at which it complies with its duties expressed through norms.

- Norms represent goals whose satisfaction may benefit the goals of other agents. Consequently, when addressees do not comply with their norms, the goals of beneficiary agents are badly affected. Thus, some goals are affected by the normative decisions of others. We say that the *expected* contributions from the society depend on the number of norms that benefit an agent and that are fulfilled by other agents.

This chapter aims to provide experimental evidence for the effects of autonomous norm compliance on both agents and their society. In particular, the three aspects above are analysed in societies comprising agents that follow different strategies for norm compliance. In this way, we observe how successful an agent is regarding the satisfaction of its goals, the contributions it gives to its society and the contributions it receives from the society. These observations also allow us to verify the effectiveness of norms according to the kind of members that comprise a society. The chapter is organised as follows. Section 8.2 describes the different strategies for norm compliance that agents can choose. Our experimental testbed is described in Section 8.3, while Section 8.4 describes both the kinds of agents and the types of societies over which experiments are performed. Section 8.5 shows the result of the experiments, and finally, our conclusions are provided.

8.2 Norm Compliance Decision-Making Strategies

Norm compliance strategies are used by agents to decide whether a norm must be rejected or complied with. This decision is based on an agent's goals and motivations, and on the normative behaviour of other agents. Our model of norm-compliance includes strategies that are classified according to the elements that agents take into account to make a normative decision. Three groups of norm-compliance strategies are considered: simple, motivated and influenced strategies, which are summarised as follows (and explained in detail in Chapter 7).

Simple Strategies Those strategies in which agents do not reflect about the effects of complying with a norm on their goals are called simple strategies. They include the following cases.

- Agents use a *social* norm-compliance strategy when norms are complied with as an end.
- Agents that reject any norm also as an end, are using a *rebellious* strategy.
- *Fearful* strategies are those in which norms are fulfilled when they include punishments of any kind.
- *Greedy* strategies are those in which norms are fulfilled when they include any kinds of rewards.

Motivated Strategies Motivated strategies for norm-compliance are those in which agents take into account the effects of fulfilling or violating a norm on their goals and motivations. Each strategy is divided into two cases depending on whether the norm conflicts with goals. A norm is conflicting if its normative goals hinder an agent's goals. The description of these strategies is given below.

- In a *pressured* strategy, agents compare the effects of complying with a norm against the effects of being penalised if the norm is violated, as follows.
  - A non-conflicting norm is complied with if there is a goal that is hindered by its punishments.

    - A conflicting norm is complied with if the goals hindered by its punishments are more important than the goals hindered by the normative goals.

- In an *opportunistic* strategy, agents compare the effects of complying with norms against the effects of losing the rewards offered if norms are not fulfilled.
    - A non-conflicting norm is complied with if some goals can benefit from its rewards.
    - A conflicting norm is complied with if the goals that can benefit from its rewards are more important than the goals hindered by normative goals.

Influenced Strategies All strategies in which, to take a decision, agents observe the normative behaviour of other agents are called influenced strategies. These are described as follows.

- An agent uses a *simple imitation* strategy when it does what other agents do. That is, if another agent complies with a norm, the agent also complies with it. Similarly, an agent rejects any norm that has been rejected by others.
- In a *reasoned imitation* strategy, agents imitate successful agents. That is, if a norm is violated by another agent and the agent is not penalised, this agent also violates the norm. Similarly, if another agent complies with the norm and is rewarded for doing so, this agent also complies with the norm.
- Agents that use a *reciprocation* strategy take into account the previous normative behaviour of the beneficiaries of a norm. That is, if the norm provides some benefits to another agent from whom help was received in the past, the agent complies with the norm.

In all cases not described, norms are just rejected. Our experiments are only focussed on simple and motivated strategies.

8.3 Experimental Testbed

8.3.1 Workbench

As a base for experimentation, we have developed a workbench in Java in which the normative behaviour of a society of autonomous agents can be simulated and observed.

The workbench provides the means to manipulate the most relevant aspects of a society so that new agents can be defined, added or eliminated from the environment, and new sets of norms, goals for agents and conflicts between norms and goals can be created at any time.

This workbench implements the agents defined in Chapter 3, the norms and normative multi-agent systems defined in Chapter 4, and the normative behaviour of agents regarding norm compliance as defined in Chapter 7. Concepts defined in Chapters 5 and 6 were neither implemented nor taken into account for experimentation, since they can be considered to be subsidiary aspects.

To be executed, an *experiment* requires a set of agents, a set of norms, a set of goals and the set of *conflicts* that represents the cases in which a norm conflicts with an agent's goal. After these parameters are fixed, a *step* is executed by selecting random subsets of goals and norms with which each agent performs its norm deliberation and norm compliance processes. Then, the results of all agents' normative decisions are gathered by the workbench. In order to avoid error, a step is executed a predetermined number of times. This sequence of steps is called a *test*. A complete experiment consists of a sequence of tests, varying the number of conflicts from no conflicts at all to 25%, 50%, 75% and 100% of conflicts.

8.3.2 Control Parameters

Before an experiment or test can be made, some parameters are fixed by the workbench and others are defined by the user. The workbench calculates the following.

- A set of goals to represent all the goals (*goalbase*) that an agent might have is generated.
- A random motivation value is assigned to each goal. This value is in the range of $[0.0, 1.0]$ where the value 1.0 represents the highest motivation.
- Each goal in the *goalbase* set is used as a normative goal to create a set of norms (*norms*). By doing this, we give the same probability to all goals of becoming hindered by a norm.
- Both *punishments* and *rewards* of each norm are randomly generated, again by using the *goalbase* set.
- The *goalbase* set is also used to generate a set of goals in conflict (*conflicts*), where each conflict is represented as a pair of goals (g_1,g_2). This means that

when g_1 represents a normative goal or a punishment, it is in conflict with g_2, where g_2 is one of an agent's current goals.

The kinds of agents in a society, the number of steps in each test, the number of conflicts in each test, and the number of goals and norms that become active at a time are arbitrarily fixed. Once all of these parameters are fixed, they are used until the end of a test or an experiment.

8.3.3 Step Execution

In all experiments, we assume that agents have similar capabilities, they are controlled by the same set of norms, and that all punishments are applied and all rewards are given. In each step of a test, the workbench prepares the following information to be distributed to all agents.

1. From the set of total goals (*goalbase*), a random subset of goals (*goals*) is taken to represent the current goals of agents.
2. A set of random norms (*newnorms*) is taken from the set of norms (*norms*).
3. Random agents are selected as beneficiaries and addressees for each norm in the *newnorms* set.
4. Each agent is required to make their decisions regarding *goals* and *newnorms*. (i.e. Norm deliberation and norm compliance processes are executed by each agent.)
5. Each agent is allowed to observe the normative decisions of other agents, to update their records regarding the norms they are expecting to be complied with by other agents.
6. Relevant information is gathered and a new step is executed.

In each step of a test, each agent performs the following activities.

1. The agent takes *goals* as the set of current goals.
2. From the *newnorms* set, the agent selects the norms addressed to it as the set of active norms (*activenorms*), and the norms from which benefits are expected (*expectednorms*).

3. The set of *activenorms* is divided into two sets of *conflicting* and *non-conflicting* norms, and *norm deliberation* strategies are applied to each of these subsets in order to decide which norms to fulfill and which norms to reject.

4. The *norm compliance* process is executed to update the set of current goals according to the previous normative decisions.

8.3.4 Data Gathering

At the end of each step, and for each agent, data is gathered as follows.

- The number of norms that are active during a step (*tactivens*), and the number of these norms that the agent complies with (*tintendedns*).
- The number of an agent's goals (*totalgs*), and the number of these goals (*tsatisfiedgs*) that were not hindered by normative causes such as conflicts with normative goals, punishments applied due to unfulfilled norms, or goals unsatisfied due to other agents not complying with norms.
- Finally, the number of norms for which the agent is a beneficiary (*texpectedns*) and the number of these norms that are fulfilled by their addressees (*texpectedcompliedns*) are also gathered.

The *AgentPerformanceRegister* schema includes variables to represent the information described above. (This and the remaining schemas in this chapter are not part of the framework developed in this thesis but they help to understand the way in which the gathered information is related.) There, the *expectednorms* and *expectedcompliednorms* variables are included. The first represents all those norms for which the agent is a beneficiary, whereas the second is the subset of these norms that are fulfilled by their addressees. The normative goals of expected norms are goals of the agent. Moreover, the *normativeAgSt* function is a function that given the name of an agent returns the corresponding model of its state.

$$
\begin{array}{|l}
\multicolumn{1}{l}{\_\_ \mathit{AgentPerformanceRegister} \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_} \\
\mathit{NormativeAgentState} \\
\mathit{society} : \mathit{NormativeMAS} \\
\mathit{expectednorms}, \mathit{expectedcompliednorms} : \mathbb{P}\,\mathit{Norm} \\
\mathit{tactivens}, \mathit{tintendedns} : \mathbb{N} \\
\mathit{totalgs}, \mathit{tsatisfiedgs} : \mathbb{N} \\
\mathit{texpectedns}, \mathit{texpectedcompliedns} : \mathbb{N} \\
\mathit{normativeAgSt} : \mathit{AgentName} \nrightarrow \mathit{NormativeAgentState} \\
\hline
\mathit{self} \in \mathit{society.members} \\
\forall\, \mathit{en} : \mathit{expectednorms} \bullet (\mathit{self} \in \mathit{en.beneficiaries} \land \\
\qquad (\exists\, \mathit{an} : \mathit{society.members} \bullet \mathit{en} \in (\mathit{normativeAgSt}\ \mathit{an}).\mathit{activenorms})) \\
\forall\, \mathit{ecn} : \mathit{expectedcompliednorms} \bullet (\mathit{ecn} \in \mathit{expectednorms} \land \\
\qquad (\exists\, \mathit{an} : \mathit{ecn.addressees} \bullet \mathit{ecn} \in (\mathit{normativeAgSt}\ \mathit{an}).\mathit{intendednorms})) \\
\mathit{normgoals}\ \mathit{expectednorms} \subseteq \mathit{goals} \\
\hline
\end{array}
$$

These values must be initialised at zero at the beginning of every test, and updated after the norm compliance process is executed, as follows.

$$
\begin{array}{|l}
\multicolumn{1}{l}{\_\_ \mathit{UpdateAgentPerformance} \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_} \\
\mathit{NormativeAgentState} \\
\Delta \mathit{AgentPerformanceRegister} \\
\hline
\mathbf{let}\ \mathit{hpgs} == \mathit{hindered}(\mathit{goals}, \mathit{punishgoals}\ \mathit{rejectednorms}) \bullet \\
\mathbf{let}\ \mathit{hngs} == \mathit{hindered}(\mathit{goals}, \mathit{normgoals}\ \mathit{intendednorms}) \bullet \\
\mathbf{let}\ \mathit{negs} == \mathit{normgoals}\ (\mathit{expectednorms} \setminus \mathit{expectedcompliednorms}) \bullet \\
\qquad (\mathit{tsatisfiedgs}' = \mathit{tsatisfiedgs} + \#(\mathit{goals} \setminus (\mathit{hpgs} \cup \mathit{hngs} \cup \mathit{negs}))) \\
\mathit{tactivens}' = \mathit{tactivens} + \#\mathit{activenorms} \\
\mathit{tintendedns}' = \mathit{tintendedns} + \#\mathit{intendednorms} \\
\mathit{totalgs}' = \mathit{totalgs} + \#\mathit{goals} \\
\mathit{texpectedns}' = \mathit{texpectedns} + \#\mathit{expectednorms} \\
\mathit{texpectedcompliedns}' = \mathit{texpectedcompliedns} + \#\mathit{expectedcompliednorms} \\
\hline
\end{array}
$$

8.3.5 Normative Evaluation Parameters

As mentioned in the introduction, three values provide important information for analysing the effects of norm compliance on a society and their members. These values are the *individual performance*, the *social contribution* and the *normative expectation* of each agent. They are defined as follows, and formalised in the schema below.

- An agent's *individual performance* (*IP*) is the number of personal satisfied goals as a proportion of its total goals over the same period.
- The *social contribution*(*SC*) of an agent is the number of times that it complies with its norms in proportion to the total number of active norms. This represents how much contribution an agent provides to its society.
- The *normative expectation* of an agent (*NE*) is calculated as the ratio between the active norms addressed to other agents and that benefit the agent, and the total of these norms that are fulfilled during a specified period of time. This value represents how much a society contributes to the satisfaction of an agent's goals.

AgentPerformance

$AgentPerformanceRegister$

$sc, ip, ne : \mathbb{N}$

$ip = tsatisfiedgs \operatorname{div} totalgs$

$sc = tintendedns \operatorname{div} tactivens$

$ne = texpectedcompliedns \operatorname{div} texpectedns$

In addition, we define *norm effectiveness* as the average of the social contribution of all agents in a society. *Norm effectiveness* can be taken as a factor of measuring the risk of collapsing for a society. A lower value in this parameter means that agents are not complying with their responsibilities and, therefore, many goals of agents are being unsatisfied. As mentioned earlier, this is a cause for agents to emigrate to other societies in which better results can be obtained.

8.4 Societies of Normative Agents

8.4.1 Normative Individuals

By combining different strategies for norm compliance, different kinds of agents are represented. Table 8.1 shows the kinds of individuals considered in this chapter's experiments. In the table, *NCN* and *CN* represent *non-conflicting* and *conflicting* norms respectively. Agents are defined in terms of the strategies they use for norms that do not conflict with their goals, and strategies for those norms that conflict.

Agent	Strategies for NCN	Strategies for CN
Social	Social	Social
Rebellious	Rebellious	Rebellious
Selfish	Pressured & Opportunistic	Pressured & Opportunistic
Social-Selfish	Social	Pressured & Opportunistic

TABLE 8.1: Normative Individuals

We choose these examples because they are the most common strategies used by agents in other related works [32, 42, 54]. The first two represent extreme behaviours, i.e. agents that always obey norms or agents that always refuse them. The third case represents selfish agents that comply with norms only if this provides them with some advantages. Social-selfish agents are those willing to obey norms only if they do not conflict with their own goals, otherwise they apply a selfish strategy. Other kinds of normative behaviours can also be represented by using our model, but some combinations appear to be unrealistic and are not considered for experimentation. For example, it is possible to define agents that comply with conflicting norms and reject non-conflicting norms, which is nonsensical.

8.4.2 Societies of Normative Agents

Now, a sequence of experiments is performed by varying the composition of agents in the society. We use the normative individuals in Table 8.1 to create the different societies we want to observe as described in Table 8.2. Only two kinds of agents are considered in each society which gives us a total of ten possible kinds of societies.

Society	Individuals Class A	Individuals Class B
1	*Rebellious*	*Rebellious*
2	*Selfish*	*Selfish*
3	*Social*	*Social*
4	*Social-Selfish*	*Social-Selfish*
5	*Social*	*Rebellious*
6	*Selfish*	*Rebellious*
7	*Social-Selfish*	*Rebellious*
8	*Selfish*	*Social*
9	*Selfish*	*Social-Selfish*
10	*Social*	*Social-Selfish*

TABLE 8.2: Societies of Normative Agents

In addition, for those societies including more than one kind of individual, experiments are performed by considering different proportions in the population of agents. Since we have only two kinds of agents in each society, these proportions are selected in order to perform an experiment in which the population of a kind of individual is predominant over the other, and an experiment in which the proportions of individuals of each kind is the same.

1. 25% individuals class A, 75% individuals class B.
2. 50% individuals class A, 50% individuals class B.
3. 75% individuals class A, 25% individuals class B.

8.5 Experimental Analysis

As explained above, an experiment consists of five tests: one in which norms do not conflict with individual agent goals, and the rest in which 25%, 50%, 75% and 100% of the norms conflict. Each test of an experiment is executed over 500 steps, where ten goals and ten norms are used. At each step, a maximum of ten norms per agent are active, and there are a maximum of three goals per agent.

The number of steps of a test was validated by using the *t-Test: Two-Sample Assuming Equal Variances* analysis, by which we ensure that the results obtained will not change with longer runs of experiments. The *t-test* thus ensures that the results are *statistically significant*. The *t-test* was performed on two samples consisting of ten sets of data of the variable to be validated. Thus, with *individual performance (IP)*, we take the data obtained by running the experiment 500 times over 10 turns to get the first sample. We do the same for the second sample, but 800 times instead of 500.

	IP(500)	IP(800)
Mean	0.415289	0.411881
Variance	5.00675E-05	4.77996E-05
Observations	10	10
Hypothesized Mean Difference	0.0	
$P(T <= t)$	0.145177442	

TABLE 8.3: t-Test: Two-Sample Assuming Equal Variances

The *significance level* used is 0.05, which gives a *confidence level* of 95%. All t-tests were performed with 95% of confidence. Table 8.3 shows the results of the *t-test* for the *IP* of a society comprising *social-selfish* agents. The important value to observe in

Table 8.3 is P which, when greater than the significance level (i.e. $P > 0.05$), means that the results are valid.

Experiments were run for ten different societies, and each with three different proportions of mixed populations. Although we performed the test on the most representative societies (i.e. those with equal percentage of mixed populations), we provide details of the statistical significance tests of only a select few, in Table 8.4, to demonstrate the process. (The number of a society is taken according to the Table 8.2.)

Society	$P(T <= t)$
S4:SocialSelfish	0.145177442
S6: Selfish & Rebellious	0.053529954
S8: Selfish & Social	0.431459786
S9: Selfish & SocialSelfish	0.066861644

TABLE 8.4: t-Test Results

The effects of norm compliance for each kind of agent are shown in graphs where the individual performance of agents is indicated by triangles (▲ *IP*), the social contribution value is indicated by small circles (• *SC*), and the normative expectation value is indicated by squares (■ *NE*).

8.5.1 Uniform Societies

The first set of experiments was performed by considering societies with similar members; that is, societies in which all their members follow the same strategies for norm compliance. This corresponds to societies 1 to 4 from Table 8.2. The results are shown in Figure 8.1, in which the *norm effectiveness* value corresponds to the social contribution of an agent.

As expected, the normative expectation of agents (■ *NE*) corresponds to their social contributions (• *SC*) because all agents use the same strategies. Now, if the individual performance of agents (▲ *IP*) in the different kinds of societies is analysed, we can observe that, compared with the other three kinds of agents, rebellious agents achieve the worst individual performance. This is due to no one in the society complying with responsibilities that might help others. Social agents achieve better performance than selfish agents. This can be explained by the fact that in societies of social agents, the goals of their members are satisfied through the responsibilities of other agents that always comply with them. In addition, when social-selfish and selfish strategies are compared, we can observe that both the individual performance and the social contribution have higher values for social-selfish than for selfish strategies. This means, that

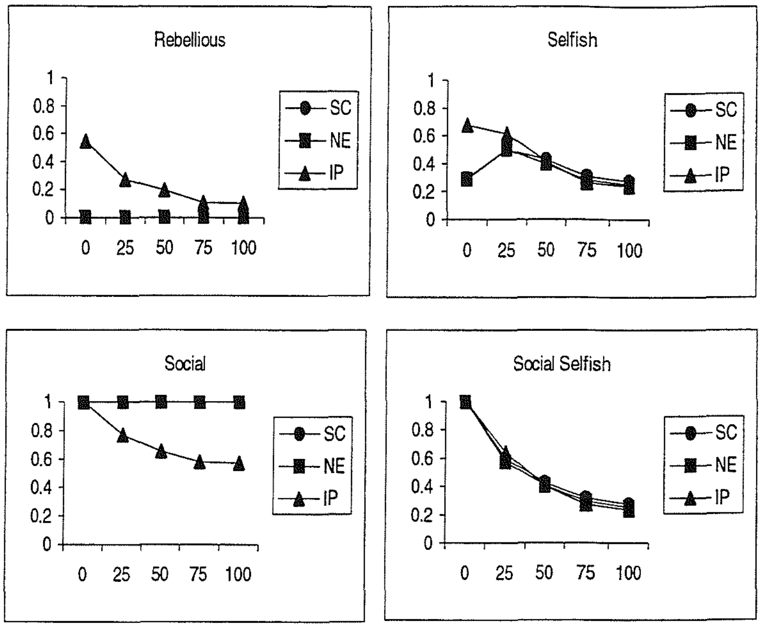


FIGURE 8.1: Societies of Similar Agents

social-selfish strategies provide better result for the performance of agents and for the effectiveness of the norms in a society.

8.5.2 Mixed Societies

The remaining experiments are grouped according to the kind of societies being observed and the proportion of the population of agents being considered. Thus, for each kind of society, three experiments were performed, and for each experiment, a graph to show the effectiveness of norms in the society is included.

Figures 8.2, 8.3 and 8.4 show the results of experiments performed with societies composed of social and rebellious agents, selfish and rebellious agents, and social-selfish and rebellious agents. These correspond to societies 5, 6, and 7 of Table 8.2, respectively. As can be seen, as soon as different kinds of agents are considered, the normative decisions of some affect the performance of the others. We can observe that, by contrast with societies in which all agents behave similarly, rebellious agents are the most successful regarding the satisfaction of their goals (▲ *IP*). Their individual performance is always higher than the performance of other individuals in the same society. That is, they exploit the responsibilities of other agents to satisfy their goals without providing anything in exchange. In addition, the higher the proportion of rebellious agents, the bigger the gap between what agents contribute to their society (● *SC*) and what they receive from their society (■ *NE*). The figures also show that, as expected, in

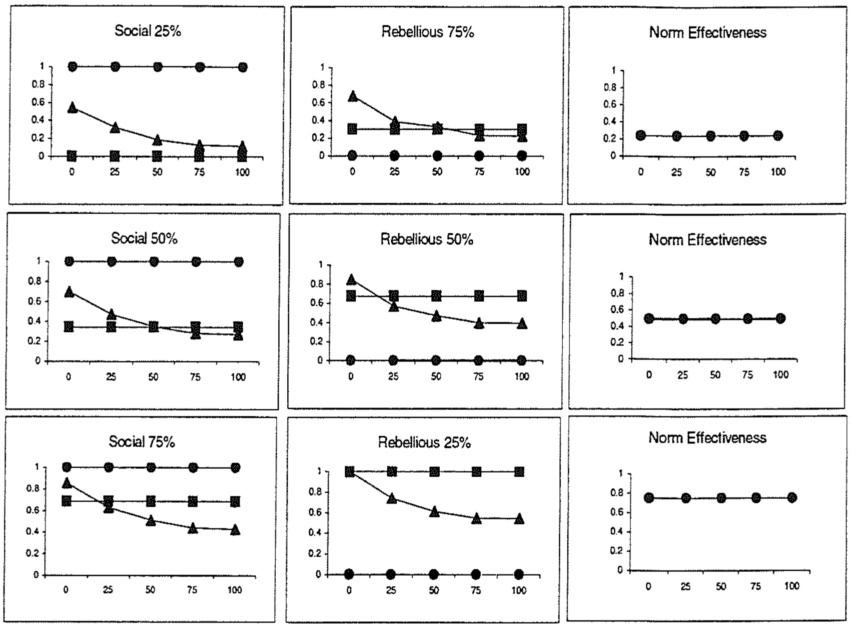


FIGURE 8.2: Societies of Social and Rebellious Agents

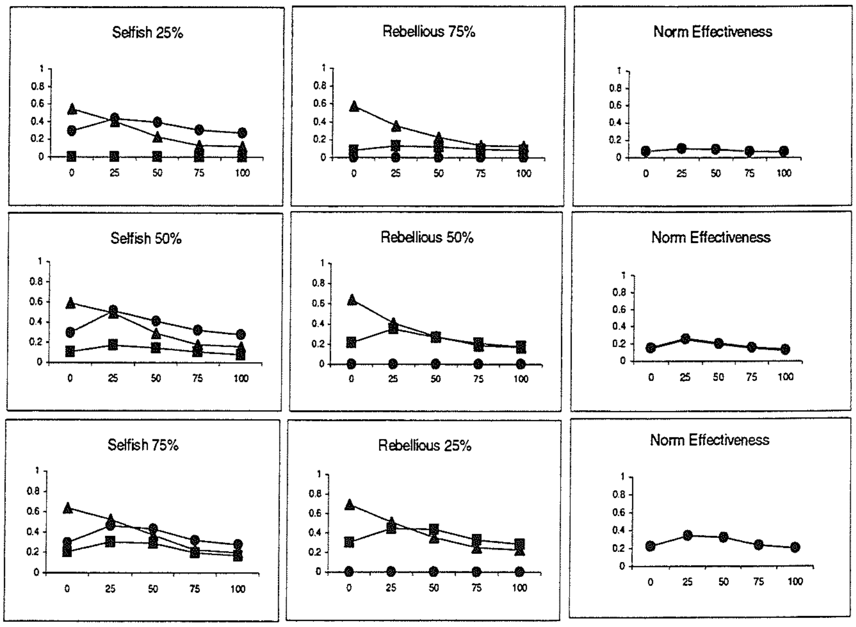


FIGURE 8.3: Societies of Selfish and Rebellious Agents

those societies with social agents, norm effectiveness is higher.

The next set of experiments is performed with societies composed of selfish and social agents, selfish and social-selfish agents, and social and social-selfish agents. These correspond to societies 8, 9, and 10 of Table 8.2. Figures 8.5, 8.6, and 8.7 show, respectively, the results of these experiments. In the three figures, we can observe that

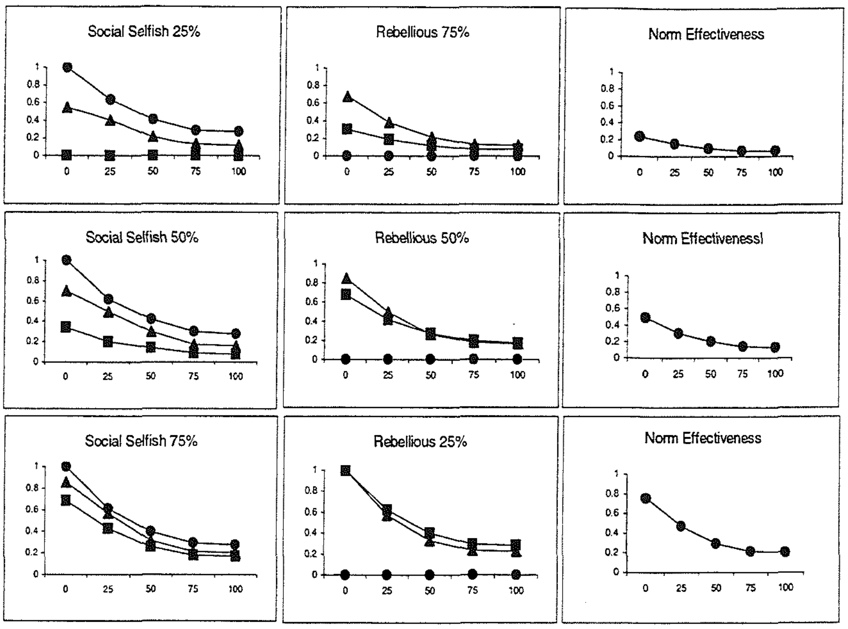


FIGURE 8.4: Societies of Social-Selfish and Rebellious Agents

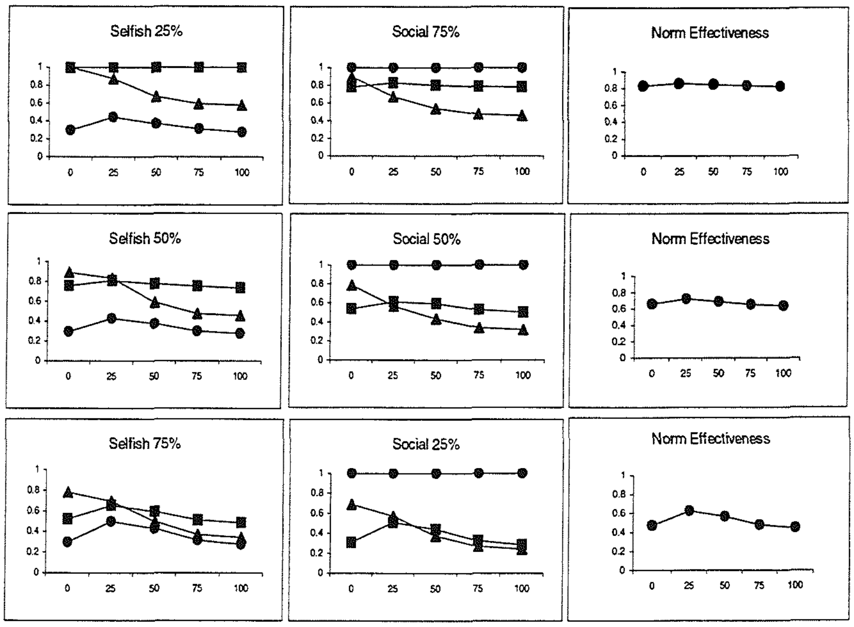


FIGURE 8.5: Societies of Selfish and Social Agents

the higher the proportion of agents following social strategies, the better the individual performance (▲ *IP*) of both kinds of agents. However, in the three cases, the individual performance of agents with a social strategy is always lower than the individual performance of agents following a selfish strategy. Then agents with a social strategy always contribute more to the satisfaction of other agents' goals (● *SC*) than they receive from

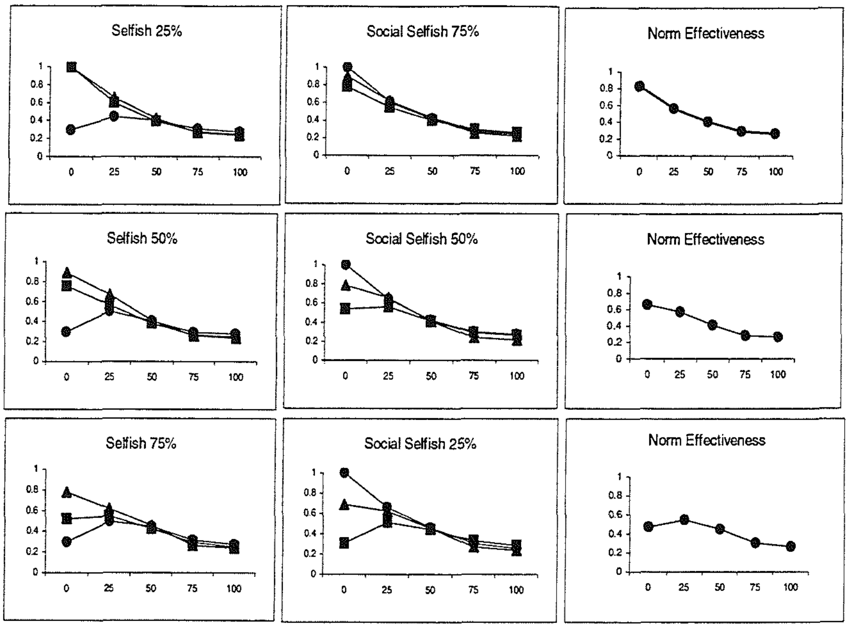


FIGURE 8.6: Societies of Selfish and Social-Selfish Agents

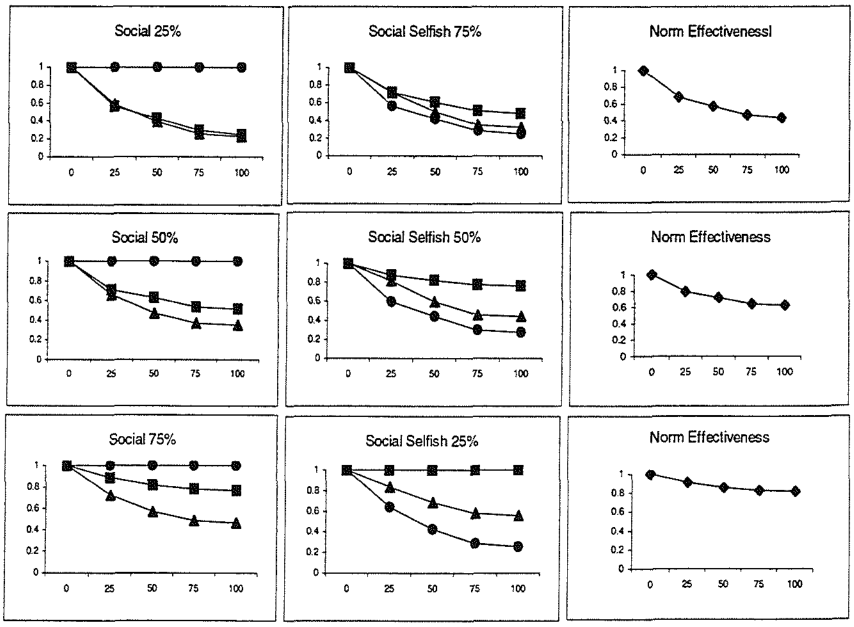


FIGURE 8.7: Societies of Social and Social-Selfish Agents

other agents (■ *NE*).

8.5.3 Discussion

From the point of view of the individual performance of agents, there are no ways to conclude that one strategy is better than another because, as was shown in the experiments, this depends on the kinds of agents with which an agent interacts. For instance, we observe that social strategies perform better only if they are used with agents following the same strategy. In all the remaining cases, agents following this strategy are always abused (i.e. other agents achieve better performance due to the help of these social agents).

Agents that follow selfish and rebellious strategies are more successful regarding the achievement of their goals when they interact with social agents. However, if the population of these kinds of agents increases, this may be counterproductive because it may cause agents that are providing more than they receive to emigrate to other societies. So, the performance of these selfish and rebellious agents might subsequently decrease.

8.6 Conclusions

In this chapter, the effects of autonomous norm compliance on societies and their members have been verified experimentally. This effects have been analysed regarding three parameters: the *individual performance*, the *social contribution*, and the *normative expectation* of agents. The individual performance refers to the number of goals that an agent can satisfy in a society where its behaviour is constrained by norms, and where the help of other agents is expected. The social contribution refers to the number of norms that an agent is willing to comply with in benefit of other agents. Finally, the normative expectation refers to the number of norms that must be complied by other agents, and whose benefits are enjoyed by the agent.

In other words, whereas individual performance gives us information about the success of an agent in the achievement of their goals, the social contribution gives information about the work the agent has done in benefit of other agents, and the normative expectation gives information about the work that others have done to benefit this agent. These parameters were evaluated in societies composed of different kinds of individuals, i.e. individuals using different strategies for deciding which norms to fulfill. In addition, the *effectiveness* of norms for each society was evaluated. This refers to the average of norms complied by all members.

The parameters defined here can be used by agents to decide which strategy better suits their interests, and to identify when they have to emigrate because either the society is too restrictive (continuous conflicting norms), or the members are too irresponsible to comply with their commitments (low effectiveness of norms). Designers of systems can also use this information to decide not only which kinds of members are preferred, but also what kinds of norms and mechanisms to enforce them must be used because non-compliance with norms may be due to that norms continuously conflict with agents' goals, or due to that punishments and rewards are not considered important for agents.

Chapter 9

Conclusions

9.1 Introduction

One of the key issues in the computational representation of *open societies* refers to the introduction of *norms* that help to cope with the heterogeneity, the autonomy and the diversity of interests among their members [122]. The work developed in this thesis addresses this issue in two main directions. It provides the means to incorporate norms into multi-agent systems, and provides the means to enable agents to reason about norms and the society in which they participate. In this chapter, the main contributions of the thesis are summarised, and the limitations of the work and topics for future research are discussed.

Our research builds on relevant work on norms and multi-agent systems such as that of Castelfranchi and Conte [21, 25, 38, 39], Jennings [93, 94] and Jones and Sergot [101, 102], as well as work on norms from a sociological, legal and philosophical view by Cohen [35], Ross [146] and Tuomela [164, 165]. In addition, the foundations of this work are taken from Luck and d'Inverno's SMART agent framework [62, 120] whose concept of *motivations* as the driving force that affects the reasoning of agents in satisfying their goals is considered as the underlying argument for agents to voluntarily comply with norms and to voluntarily enter and remain in a society.

9.2 Contributions

In what concerns the introduction of norms into multi-agent systems, and the development of autonomous agents able to participate in normative societies, the work of this thesis provides three main contributions to the fields of agents and multi-agent systems.

- This thesis provides a *unifying normative framework for agent-based systems*, and includes models that facilitate the understanding and the computational implementation of norms and multi-agent systems regulated by norms.

- It provides an extensive analysis of the *powers of autonomous agents* that are taken as the means to influence other autonomous agents which they have to interact with. This analysis serves to model other decision-making processes such as goal delegation, goal adoption and plan selection, in which powers are taken into account for effective satisfaction of goals.

- It provides the means to incorporate *decision-making processes regarding norms* into a coherent foundational model of agents in order to enable autonomous agents to represent norms, and to adopt and comply with them. This is done not as an end, but as the result of deliberation processes in which goals and motivations are taken into account.

These contributions facilitate the development of agents with characteristics that have not generally been considered previously. In particular,

- Since agents decide on the basis of their own goals and motivations whether to adopt and comply with norms, agents that voluntarily participate in a society regulated by norms can be modelled.

- Since agents can adopt new norms, agents become able to play different roles (prescribed by norms) and, consequently, can have different responsibilities in a society.

- Since agents are able to identify powers, to use them for influencing others, and to allow themselves to be influenced by others, agents that voluntarily become engaged in a relation of cooperation can be modelled.

Agents of this kind are needed for the development of dynamic and open norm-based systems whose norms might change at run-time, and whose members are not known in advance, such as electronic institutions, virtual organisations, and coalitions. In the following subsections, the particular contributions of this thesis are summarised.

9.2.1 The Normative Framework

By contrast with other frameworks in which norms are reduced to obligatory *ends* with penalties in case of violation, our framework is oriented to providing the means for agents to reason about norms and the society in which they participate. This framework includes the following.

- *A unifying model of norms that subsumes previous models and that includes elements to enable agents to reason about norms.*

  In the framework, norms are defined as mechanisms to influence the behaviour of agents with three properties: *prescriptiveness*, *sociality*, and *social pressure*. That is, norms prescribe how an agent must behave in situations in which more than one agent is involved and, since it is always expected that norms may conflict with the individual goals of agents, mechanisms to enforce compliance with norms are needed. Our general model of norms enables agents to identify the goals prescribed by the norm, the situations in which these normative goals must be satisfied, the agents that are responsible for the satisfaction of these goals, the agents that might benefit from them, and what may happen in cases in which norms are either fulfilled or violated. The model also facilitates the representation of different categories of norms ranging from *obligations* and *prohibitions*, to *social commitments* and *social codes*. Thus, instead of proposing a different model for each category of norm, they are considered as variants of one general model.

- *The means to represent a complete system of norms.*

  In systems regulated by norms, it is common to find that compliance with norms may cause other norms to become active, such as those norms aimed at punishing non-compliance or rewarding compliance with other norms. In the framework these norms are called *interlocking norms*. Interlocking norms provide a way to represent how the behaviour of the addressee of a norm might influence the behaviour of the addressee of another norm if these norms are related.

- *A model of normative multi-agent systems that acknowledges that members of a system are autonomous and, therefore, that their normative behaviour can be expected but never assured.*

  Normative multi-agent systems are defined as systems of autonomous agents whose behaviour is regulated by norms. These systems have three characteristics: *membership*, *social pressure*, and *dynamism*. Thus, agents recognise themselves as members by adopting the norms of the society and by accepting its authorities and its means of exerting social pressure, which is needed to persuade agents to comply with norms. The norms of a system include norms that entitle agents to legislate, to apply punishments and to give rewards. These norms are the means for agents to identify the *authorities* of the system (which correspond to the addressees of any of these kinds of norms) and the *limits* of their authority (because these norms specify which actions these authorities are entitled to perform). Through these norms, *order* in a system can be reached because the authority and responsibilities of agents are well defined.

- *A dynamic vision of norms.*

  Norms are not a static concept. Their inclusion in a system influences the behaviour of those agents responsible for complying with them, those agents that benefit from them, and those agents responsible for monitoring the normative behaviour of others. From the normative behaviour of the involved agents, different states of norms, which can be used by agents to take decisions, are identified. Thus, norms can be issued, adopted, rejected, complied with, violated, and their compliance can be rewarded and their violation can be penalised.

9.2.2 Agent Powers

In satisfying goals, the identification of the *power* of agents is crucial because it represents the means to influence autonomous agents. However, powers are constrained and agents must be able to identify when, and over whom, these powers can be exerted. This thesis addresses this concern by providing the following.

- *Mechanisms to identify empowered situations.*

  Two kinds of power have been identified here: *circumstantial powers* and *institutional powers*. Circumstantial powers are the powers of agents due to their own abilities and to the use of such abilities to satisfy or hinder the goals of other agents. In addition, *institutional powers* are the powers of agents due to the norms that control the society in which they participate.

- *Ways for agents to use powers to satisfy goals.*

  In our model, agents have been provided with the means to use their empowered situations to delegate their goals to other agents. That is, to choose a partner to ask for help, agents examine their relations of power in order to foresee how to satisfy a requirement for goal adoption because agents on which power can be exerted are more likely to be influenced to adopt the goals of empowered agents. Agents that correctly identify their powers can even be prepared to argue with those agents that reject the adoption of their goals. Conversely, agents have also been endowed with the means to recognise the power of others and, therefore, to recognise their liability of being influenced to adopt the goals of an empowered agent.

A classification of plans that helps agents to decide between alternatives to satisfy a goal has also been provided. That is, agents that can identify those agents liable to be influenced to adopt their goals can also make a better selection of plans. If powers can be exerted over agents that are needed to perform a plan, the plan is more likely to succeed. A criterion to select a plan can be based on both the possibility of delegating all the subgoals included in the plan and the kinds of agents to which the subgoals can be delegated.

9.2.3 Normative Behaviour

Instead of defining a new model to represent normative agents, new decision-making *processes* regarding norms have been incorporated into a coherent and foundational model of agents in order to facilitate their implementation. Regarding this concern, this thesis makes the following contributions.

- *The means for agents to voluntarily decide whether participating in a society regulated by norms will serve their own interests have been given.*

  To enter a society, an agent considers the responsibilities it acquires through norms, and the contributions that it receives from the society that can be used for the satisfaction of its goals. Thus, if social contributions allow an agent to satisfy its goals, the agent would be willing to comply with norms that might benefit other agents in the society.

  Our work has provided the means for agents to decide whether remaining in a society is important, and whether leaving a society is necessary. We have explained that, once an agent has entered a society, the satisfaction of its goals is not the only reason it remains there. There are certain relationships that are created during social interactions that motivate an agent to stay in the society.

- *Two new processes of reasoning about norms, not considered elsewhere, have been introduced.*

  Adoption of, and compliance with norms are processes that concern the addressees of a norm. They are executed at different times, and their outcomes can be contradictory. The addressees of a norm first reflect on the reasons they have to acquire the new responsibilities that an adopted norm would imply and then, during run-time, they reflect on the reasons they have to comply with this norm.

Thus, whilst norm adoption involves an agent's acknowledgment of responsibilities, norm compliance involves an agent's decision to fulfill a norm already adopted. During the norm adoption process, an internal representation of the norm is created, and during the norm compliance process, normative goals are derived from the norm and they are intended [116]. In the norm adoption process, norms are recognised as something that ought to be done and, most of the time, agents are willing to fulfill them. However, at run-time, the state of agents may change, making it difficult to satisfy the norm, especially if such a norm causes a conflict with other goals that agents consider important.

- *The means for agents to decide whether to adopt a new norm and whether to comply with an adopted norm have been provided.*

  Norm adoption is defined as the process through which agents decide to adopt a norm and, therefore, to create its corresponding internal representation. A set of norms is adopted as soon as an agent decides to enter a society, and new norms are adopted during the time the agent remains there. In contrast to other models that consider that any norm issued by an authority must be adopted, our model states that, for a norm to be adopted, the power of the issuer must be recognised. This includes cases of authorities with power as well as cases of powers acquired by other means.

  Since the situations of agents might change from the time a norm is adopted to the time it must be complied with, agents are provided with the means to decide whether to comply with a norm according to their own goals and motivations. Autonomous norm compliance has been divided into two different and related processes: the norm deliberation and norm compliance processes. Through a norm deliberation process, agents autonomously decide if a norm must be complied with or if the norm must be rejected. To do this, agents use different strategies.

  Now, through the norm compliance process, the goals of an agent are updated according to the normative decisions it has made. Thus, normative goals of intended norms are added to the set of goals, and goals satisfied through rewards are eliminated, as are those hindered by normative goals and by punishments. It is important to mention that our models are guided by the principle that *neither punishments nor rewards can be used as enforcing mechanisms if they do not affect an agent's goals*. In other words, no punishment is effective if none of an agent's goals is hindered by it, and no reward is useful if no goal can benefit from it. In addition, since goals can be affected in complying with norms, the preferences of agents over these goals are considered as the key to making any decision regarding norms.

9.3 Complexity

The analysis of the different aspects of normative agents and their reasoning processes have provided a framework for structuring the development of specific implementations or other instantiations. At the level of the framework, there are several points where, without further refinement, issues of computational complexity can give rise to intractable models and solutions. Although we don't prescribe particular ways to address these issues, we recognise the constraints they impose and outline the key issues below.

First, we note that some of the foundational concepts on which the framework is based could become computational bottlenecks. In particular, the analysis of agent systems to determine beneficial or conflicting relationships between goals (as in Section 3.4.4) relies on the use of the *logicalconsequence* predicate (also used for determining when a goal is satisfied). As indicated in Chapter 3, in the most general case, logical consequence is intractable because of the unconstrained nature of logical inference. However, this is a known and understood problem (that affects the most common BDI systems such as dMARS and AgentSpeak(L)) [58, 59] and, at the level of implementation, many possible constraints may be used to avoid combinatorial explosion. Most commonly, this can be refined to amount to simple pattern-matching or resolution, without use of extra inference rules [147]. Different implementations may take different approaches.

A second related aspect of this is in identifying which pairwise goal relationships need to be considered. If all relationships between all goals of all agents are considered, then the number of operations will become prohibitive in all but the most simple scenarios. It is clearly necessary to find ways to constrain the number of such relationships analysed, such as by pre-filtering relevant agents and goals. For any individual agent, however, the complexity depends on the ways in which it reasons about its environment.

In the case of strategies for norm compliance, agents compare their goals with the normative goals of the system rather than with other agents. Thus the kind of reasoning undertaken is limited and *can* be computationally effective.

In the case of determining the *powers* of agents, this is more problematic. A pairwise comparison of all goals with all agents in the society will not generally be possible in scenarios with more than a few agents. The solution in this case might be to pre-select a subset of agents to reason about, or to reason only about those agents identified as relevant through prior experience or through some environmental cues. The key point to note is that the work described in this thesis provides the general framework that must still be constrained further in such a way to provide an effective implementation in this particular aspect.

9.4 Limitations

The strengths of our work rely on the appropriateness of the provided models that yield computational implementations of norms, normative multi-agent systems, and normative agents. However, not all the situations or cases could be covered due to time limitations and the complexity of some aspects. In particular, our models cannot be applied in the following cases.

- *Adherence to norms.*

  We have explained the reasons for agents to adopt new norms. We have also explained that these norms are obeyed as long as agents want to remain in a society in which their goals are being satisfied. Consequently, when agents leave a society these norms no longer have any reasons to be complied with. However, some agents adhere to norms and they continue complying with them although they no longer participate in the society where the norms were adopted. Our models do not allow this situation to occur.

- *Choosing between alternative societies.*

  In our work, agents can decide when participating in a society is important for the achievement of their goals, but they can neither decide between alternative societies nor take a decision when no information about enforcement mechanisms and their responsibilities is provided.

- *Choosing the right strategy for a particular norm.*

  Our work provides models for the most common strategies for norm compliance. It also experimentally describes how the use of a particular strategy can affect both the performance of an agent and the effectiveness of the norms of a society. However, it does not say anything about which strategy must be used for particular norms. For instance, it is generally observed that norms that can be beneficial for the society, and that neither include punishments nor rewards, are complied with as an end (i.e. by using the *social strategy*). Our model neither covers the identification of these kinds of situations nor provides the means to choose between these different strategies.

9.5 Future Research

In our work, a wide range of issues concerning norms, normative agents and normative multi-agent systems has been tackled, and yet some extensions are possible. Moreover, there are other issues that were not faced in this thesis, due to the bounds imposed on the research, and that still need to be addressed. This section summarises the most important possibilities of future research.

- *Refinement of the strategies for norm compliance.*

  Our work provides nine of the most common strategies for norm compliance, but refinements of these strategies are possible. For example, the *simple imitation* strategy can be refined to imitate the behaviour of the majority of the addressees of a norm instead of imitating the behaviour of just one. The *reasoned imitation* strategy can be refined to include the probabilities of being punished by defenders of a norm, and the *reciprocation* strategy might consider the importance of the benefits received in reciprocating the past actions of agents.

- *Evaluation of alternative societies.*

  We have provided the means for agents to decide when entering and remaining in a society is important. This decision is made on the basis of the responsibilities that agents will have and the contributions to the satisfaction of their goals that agents will receive from the society. However, when agents have the opportunity to choose between more than one society to satisfy their goals, other criteria must be considered to make such an important decision. More research must be done regarding this issue.

- *Creation of norms.*

  An important topic of research concerns the creation of norms. Current research addresses this problem from the view of the *emergence* of norms between a group of agents which, through constant interaction, achieve a standard behaviour (considered as a norm) that is beneficial for all the members. However, when large and very well structured societies of agents are considered, the creation of norms relies on a group of these agents which, among other things, must analyse and identify under which conditions a new norm must be created, the purpose of the norm, towards which agents the norm must be addressed, and the means for the norm to be distributed among addressee agents.

- *Learning of norms.*

In this work, the adoption of norms has been taken as the internalisation of an issued norm. However, some norms are *learned* from interactions with other agents rather than being *issued*. Research must be done to explain in which situations the behaviour of others can be considered as a norm, and to provide the means for agents to identify the components of a norm to internalise it.

- *Experimental work on norms.*

  In this thesis, initial experiments to observe the impact of norm compliance on agents and societies were performed. Through the results, conclusions about which strategy provides better results for agents and which kind of agents are preferred to maintain order in a society were made. However, only a limited subset of strategies and limited populations of agents were considered for experimentation. Experimental work to overcome these limitations must be done.

9.6 Concluding Remarks

Our work contributes to research on norms in agents and multi-agent systems in providing answers to some of the questions that have concerned researchers on norms [44, 45]. Some have argued that autonomy and being constrained by norms are irreconcilable concepts. In our work, we have shown how agents can be autonomous and still adopt and comply with norms if, by doing so, their goals can be satisfied. In this work, the concepts of autonomy and motivations play a key role. We have given the means to represent norms and systems regulated by norms, and we have explained their roles in controlling the behaviour of autonomous agents. We have also introduced reasoning capabilities regarding norms in autonomous agents by providing the way in which norms are processed inside an agent, and the way in which other processes of reasoning are affected by them.

Our research contributes to a better understanding of norms and the role they play in enabling heterogeneous and autonomous agents to interact each other. It also contributes to the computational representation of agents which besides having properties such as flexibility to act in dynamic environments, have abilities to participate in a society while their goals are being satisfied. These contributions are a step towards the computational representation of *open societies* such as electronic institutions, virtual organisations and coalitions of agents, whose members do not necessarily share the same interests; they do not know and might not trust each other, but can work together and help each other. Such systems are likely to become prevalent in the next generation of agent based computing. The contributions of this thesis are vital in taking us further down the path towards that goal.

Bibliography

[1] P. Agre and D. Chapman. Pengi: An implementation of a theory of activity. In *Proceedings of the Sixth National Conference of Artificial Intelligence*, pages 268–272. AAAI Press/MIT Press, 1987.

[2] E. Alonso. How individuals negotiate societies. In Y. Demazeau, editor, *Proceedings of the Third International Conference on Multi-Agent Systems (ICMAS'98)*, pages 18–25. IEEE Computer Society Press, 1998.

[3] E. Alonso. Rights for multi-agent systems. In M. d'Inverno, M. Luck, M. Fisher, and C. Preist, editors, *Foundations and Applications of Multi-Agent Systems (UK-MAS 1996-2000)*, LNAI 2403, pages 59–72. Springer-Verlag, 2002.

[4] R. Arkin and T. Balch. AuRA: Principles and practice in review. *Journal of Experimental and Theoretical Artificial Intelligence*, 9(2):175–189, 1997.

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

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

[7] M. Barbuceanu and M. Fox. Coordinating multiple agents in the supply chain. In *Proceedings of the 5th International Workshops on Enabling Technologies: Infrastructure for Collaborative Enterprises (WET ICE'96)*, pages 134–142. IEEE Computer Society Press, 1996.

[8] M. Barbuceanu, T. Gray, and S. Mankovski. Coordinating with obligations. In K. Sycara and M. Wooldridge, editors, *Proceedings of the Second International Conference on Autonomous Agents*, pages 62–69. ACM Press, 1998.

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

[10] M. Beer, M. d'Inverno, N. Jennings, M. Luck, C. Preist, and M. Schroeder. Negotiation in multi-agent systems. *The Knowledge Engineering Review*, 14(3):285–289, 1999.

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

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

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

[14] A. Bond and L. Gasser. An analysis of problems and research in DAI. In A. Bond and L. Gasser, editors, *Readings in Distributed Artificial Intelligence*, pages 3–35. Morgan Kaufmann Publisher Inc., 1988.

[15] M. Bratman, D. Israel, and M. Pollack. Plans and resource bounded practical reasoning. *Computational Intelligence*, 4(4):349 -353, 1988.

[16] M.E. Bratman. What is intention? In P.R. Cohen, J.L. Morgan, and M.E. Pollack, editors, *Intentions in Communication*, pages 15–32. MIT Press, 1990.

[17] W. Briggs and D. Cook. Flexible social laws. In C. Mellish, editor, *Proceedings of the 14th International Joint Conference on Artificial Intelligence*, pages 688–693. Morgan Koufman, 1995.

[18] R. Brooks. Intelligence without representation. *Artificial Intelligence*, 47:139–159, 1991.

[19] B. Burmeister and K. Sundermeyer. Cooperative problem-solving guided by intentions and perceptions. In E. Werner and Y. Demazeau, editors, *Decentralized A.I. 3*, pages 77–92. North Holland Publishers, 1992.

[20] K. Carley and L. Gasser. Computational organization theory. In G. Weiss, editor, *Multi-Agent Systems: A Modern Approach to Distributed Artificial Intelligence*, pages 299–330. MIT Press, 1999.

[21] C. Castelfranchi. Social power. A point missed in multi-agent, DAI and HCI. In Y. Demazeau and J.P. Müller, editors, *Decentralized A.I.*, pages 49–62. Elsevier Science, 1990.

[22] C. Castelfranchi. Commitments: From individual intentions to groups and organizations. In V. Lesser and L. Gasser, editors, *Proceedings of the First International Conference on Multi-Agent Systems (ICMAS'95)*, pages 186–196. AAAI Press/MIT Press, 1995.

[23] C. Castelfranchi. Guarantees for autonomy in cognitive agent architecture. In M. Wooldridge and N. Jennings, editors, *Intelligent Agents (ATAL'94)*, LNAI 890, pages 56–70. Springer-Verlag, 1995.

[24] C. Castelfranchi. Modelling social action for AI agents. *Artificial Intelligence*, 103:157–182, 1998.

[25] C. Castelfranchi. Prescribed mental attitudes in goal adoption and norm adoption. *Artificial Intelligence and Law*, 7(1):37–50, 1999.

[26] C. Castelfranchi. All I understand about power (and something more). Technical report, ALFEBIITE Project, London, 2000.

[27] C. Castelfranchi and R. Conte. Distributed artificial intelligence and social science: Critical issues. In G. O'Hare and N. R. Jennings, editors, *Foundations of Distributed Artificial Intelligence*, pages 527–542. John Wiley & Sons, 1996.

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

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

[30] C. Castelfranchi and R. Falcone. Delegation conflicts. In M. Boman and W. Van de Velde, editors, *Multi-Agent Rationality (MAAMAW'97)*, LNAI 1237, pages 234–254. Springer-Verlag, 1997.

[31] C. Castelfranchi, M. Miceli, and A. Cesta. Dependence relations among autonomous agents. In E. Werner and Y. Demazeau, editors, *Decentralized A.I. 3*, pages 215–231. North Holland Publishers, 1992.

[32] A. Cesta, M. Miceli, and P. Rizzo. Help under risky conditions: Robustness of the social attitude and system performance. In V. Lesser and L. Gasser, editors, *Proceedings of the First International Conference on Multi-Agent Systems (ICMAS'95)*, pages 18–25. AAAI Press/MIT Press, 1995.

[33] E. Clarke and J. Wing. Formal methods: State of the art and future directions. *ACM Computing Surveys*, 28(4):626–643, 1996.

[34] P.R. Cohen and H. Levesque. Intention is choice with commitment. *Artificial Intelligence*, 42:213–261, 1990.

[35] P.S. Cohen. *Modern Social Theory.* Heinemann, 1968.

[36] R. Conte. Social intelligence among autonomous agents. *Computational & Mathematical Organization Theory*, 5(3):203–228, 1999.

[37] R. Conte. Emergent (info)institutions. *Journal of Cognitive Systems Research*, 2:97–110, 2001.

[38] R. Conte and C. Castelfranchi. *Cognitive and Social Action.* UCL Press, 1995.

[39] R. Conte and C. Castelfranchi. Norms as mental objects. From normative beliefs to normative goals. In C. Castelfranchi and J. P. Müller, editors, *From Reaction to Cognition (MAAMAW'93)*, LNAI 957, pages 186–196. Springer-Verlag, 1995.

[40] R. Conte and C. Castelfranchi. Simulating multi-agent interdependencies. A two-way approach to the micro-macro link. In K. Troitzsch, U. Mueller, N. Gilbert, and J. E. Doran, editors, *Social Science Microsimulation*, pages 394–415. Springer-Verlag, 1998.

[41] R. Conte and C. Castelfranchi. From conventions to prescriptions. towards an integrated view of norms. *Artificial Intelligence and Law*, 7(4):323–340, 1999.

[42] R. Conte and C. Castelfranchi. Are incentives good enough to achieve (info)social order? In C. Dellarocas and R Conte, editors, *Social Order in Multi-Agent Systems*, pages 45–61. Kluwer Academic Publishers, 2001.

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

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

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

[46] R. Conte and J. Sichman. DEPNET: How to benefit from social dependence. *Journal of Mathematical Sociology*, 20(2-3):161–177, 1995.

[47] K. Cook and M. Emerson. Power, equity and commitment in exchange network. *American Sociological Review*, 43(5):721–739, 1978.

[48] A. da Rocha Costa, J. Hübner, and R. Bordini. On entering an open society. In *The XI Brazilian Symposium on Artificial Intelligence*, pages 535–546. Brazilian Computer Society, 1994.

[49] M. Dastani, V. Dignum, and F. Dignum. Organizations and normative agents. In M. Shafazand and A. Tjoa, editors, *EurAsia-ICT 2002: Information and Communication Technology*, LNCS 2510, pages 982–989, 2002.

[50] N. David, J. Sichman, and H. Coelho. Agent-based social simulation with coalitions in social reasoning. In S. Moss and P. Davidsson, editors, *Proceedings of the Second International Workshop on Multi-Agent Based Simulation (MABS'00)*, LNAI 1979, pages 244–265. Springer-Verlag, 2000.

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

[52] F. Dignum. Autonomous agents and social norms. In R. Falcone and R. Conte, editors, *Proceedings of the Workshop on Norms, Obligations and Conventions at ICMAS'96*, pages 56–71, 1996.

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

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

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

[56] M. d'Inverno. *Agents, Agency and Autonomy: A formal Computational Model.* PhD thesis, University of London, England, 1998.

[57] M. d'Inverno, M. Fisher, A. Lomuscio, M. Luck, M. de Rijke, M. Ryan, and M. Wooldridge. Formalisms for multi-agent systems. *The Knowledge Engineering Review*, 12(3):315–321, 1997.

[58] M. d'Inverno, K. Kinny, M. Luck, and Wooldridge M. A formal specification of dMARS. In M. Singh, A. Rao, and M. Wooldridge, editors, *Intelligent Agents IV (ATAL'97)*, LNAI 1365, pages 155–176. Springer-Verlag, 1998.

[59] M. d'Inverno and M. Luck. Engineering AgentSpeak(L): A formal computational model. *Journal of Logic and Computation*, 8(3):233–260, 1988.

[60] M. d'Inverno and M. Luck. A formal view of social dependence networks. In C. Zhang and D. Lukose, editors, *Proceedings of the First Australian Workshop on DAI*, LNAI 1087, pages 115–129. Springer-Verlag, 1996.

[61] M. d'Inverno and M. Luck. Understanding autonomous interaction. In W. Wahlster, editor, *Proceedings of the 12th European Conference on Artificial Intelligence(ECAI'96)*, pages 529–536. John Wiley & Sons, 1996.

[62] M. d'Inverno and M. Luck. *Understanding Agent Systems*. Springer-Verlag, 2001.

[63] M. d'Inverno, M. Luck, and M. Wooldridge. Cooperation structures. In *Proceedings of the Fifteenth International Joint Conference on Artifical Intelligence (IJCAI'97)*, pages 600–605, 1997.

[64] J. Doran, S. Franklin, N. Jennings, and T. Norman. On cooperation in multi-agent systems. *The Knowledge Engineering Review*, 12(3):309–314, 1997.

[65] E. Durfee. Planning in distributed artificial intelligence. In G. O'Hare and N. R. Jennings, editors, *Foundations of Distributed Artificial Intelligence*, pages 231–245. John Wiley & Sons, 1996.

[66] E. Durfee, V. Lesser, and D. Corkill. Coherent cooperation among communicating problem solvers. *IEEE Transactions on Computers*, C-36(11):1275–1291, 1987.

[67] E. Durfee, V. Lesser, and D. Corkill. Partial global planning: A coordination framework for distributed hyphotesis formation. *IEEE Transactions on Systems Man and Cybernetic*, 21(5):1167–1183, 1991.

[68] M. Esteva, D. de la Cruz, and C Sierra. ISLANDER: an electronic institutions editor. In C. Castelfranchi and W.L. Johnson, editors, *Proceedings of The First*

International Joint Conference on Autonomous Agents and Multi Agent Systems AAMAS'02, pages 1045–1052. ACM Press, 2002.

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

[70] M. Esteva, J. Rodriguez-Aguilar, J. Arcos, C. Sierra, and P. Garcia. Formalising agent mediated electronic institutions. In F. Dignum and C. Sierra, editors, *Agent Mediated Electronic Commerce*, LNAI 1991, pages 126–147. Springer-Verlag, 2001.

[71] A. Faraizi. *Understanding Social Life*. Distance and Flexible Learning Centre. Central Queensland University, 2001.

[72] P. Faratin, C. Sierra, and N. Jennings. Negotiation decision functions for autonomous agents. *Journal of Robotics and Autonomous Systems*, 24(3-4):159–182, 1998.

[73] E. Fehr and S. Gächter. Cooperation and punishment in public goods experiments. *American Economic Review*, 9(4):980–994, 2000.

[74] R. Fennell and V. Lesser. Parallelism in artificial intelligence problem solving: A case study of Hearsay II. *IEEE Transactions on Computers*, C-26(2):106–119, 1977.

[75] IA. Ferguson. Towards an architecture for adaptive, rational, mobile agents. In E. Werner and Y. Demazeau, editors, *Decentralized A.I. 3*, pages 249–262. North Holland Publishers, 1992.

[76] K. Fischer, J. P. Müller, and M. Pischel. Unifying control in a layered agent architecture. Technical Report TM-94-05, Deutsches Forschungszentrum für Künstliche Intelligenz, Germany, 1994.

[77] M. Fox. An organizational view of distributed system. *IEEE Transactions on Systems, Man and Cybernetics*, SMC-11(11):70–80, 1981.

[78] J. French and B. Raven. The bases of social power. In D. P. Cartwright, editor, *Studies in Social Power*, pages 150–167. The University of Michigan, 1959.

[79] M. Genesereth, M. Ginsberg, and J. Rosenschein. Cooperation without communication. In *Proceedings of the Fifth National Conference of Artificial Intelligence*, pages 561–567. AAAI Press/MIT Press, 1986.

[80] M. Georgeff and A. Lansky. Reactive reasoning and planning. In *Proceedings of the Sixth National Conference of Artificial Intelligence*, pages 677–682. AAAI Press/MIT Press, 1987.

[81] R. Goodwin. Formalizing properties of agents. Technical Report CMU-CS-93-159, Carnegie Mellon University, Pittsburgh, PA 15213, 1993.

[82] A. Gouldner. The norm of reciprocity: A preliminar statement. *American Sociological Review*, 25(2):161–178, 1960.

[83] N. Griffiths. *Motivated Cooperation in Autonomous Agents*. PhD thesis, University of Warwick, England, 2000.

[84] N. Griffiths and M. Luck. Cooperative plan selection through trust. In F. Garijo and M. Boman, editors, *Multi-Agent System Engineering (MAAMAW'99)*, LNAI 1647, pages 115–129. Springer-Verlag, 1999.

[85] B. Grosz, S. Kraus, D. Sullivan, and Das S. The influence of social norms and social consciousness on intention reconciliation. *Artificial Intelligence*, 142(2):147–177, 2002.

[86] A. Haddadi and Sundermeyer. Belief-desire-intention agent architectures. In G. O'Hare and N. R. Jennings, editors, *Foundations of Distributed Artificial Intelligence*, pages 169–185. John Wiley & Sons, 1996.

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

[88] B. Hayes-Roth and J. Larsson. A domain-specific software architecture for a class of intelligent patient monitoring agents. *Journal of Theoretical and Experimental AI*, 8(2):149–171, 1996.

[89] C. Hewitt. Offices are open systems. *ACM Transactions on Office Information Systems*, 4(3):271–287, 1986.

[90] H. Hexmoor. A cognitive model of situated autonomy. In R. Kowalczyk, S. Loke, N. Reed, and G. Williams, editors, *Advances in Artificial Intelligence*, LNAI 2112, pages 325–334. Springer-Verlag, 2001.

[91] L. Hogg and N. Jenning. Socially intelligent reasoning for autonomous agents. *IEEE Transactions on Systems, Man, and Cybernetics. Part A: Sytems and Humans*, 31(5):381–393, 2001.

[92] F. Ingrand, M. Georgeff, and A. Rao. An architecture for real-time reasoning and system control. *IEEE Expert: Intelligent Systems and Their Applications*, 7(6):34–44, 1992.

[93] N. Jennings. On being responsible. In E. Werner and Y. Demazeau, editors, *Decentralized A.I. 3*, pages 93–102. North Holland Publishers, 1992.

[94] N. Jennings. Commitments and conventions: The foundation of coordination in multi-agent systems. *The Knowledge Engineering Review*, 8(3):223–250, 1993.

[95] N. Jennings. Specification and implementation of a belief-desire-joint-intention architecture for collaborative problem solving. *International Journal of Intelligent and Cooperative Information System*, 2(3):289–318, 1993.

[96] N. Jennings and J. Campos. Towards a social level characterisation of socially responsible agents. *IEE Proceedings on Software Engineering*, 144(1):11–25, 1997.

[97] N. Jennings, M. Mamdani, I. Laresgoiti, J. Perez, and J. Corera. GRATE: A general framework for cooperative problem solving. *IEE-BCS Journal of Intelligent Systems Engineering*, 1(2):102–104, 1992.

[98] N. Jennings, T. Norman, and P. Faratin. ADEPT: An agent-based approach to business process management. *ACM SIGMOD*, 27(4):32–39, 1998.

[99] N.R Jennings, J. M. Corera, L. Laresgoiti, H. E. Mamdani, F. Perriollat, P. Skarek, and L. Z. Varga. Using ARCHON to develop real-world DAI applications for electricity transportation management and particle accelerator control. *IEEE Expert*, 11(6):60–88, 1995.

[100] A. Jones and M. Sergot. Deontic logic in the representation of law: Towards a methodology. *Artificial Intelligence and Law*, 1(1):45–64, 1992.

[101] A. Jones and M. Sergot. On the characterisation of law and computer systems: The normative systems perspective. In J. Meyer and R. Wieringa, editors, *Deontic Logic in Computer Science: Normative System Specification*, pages 275–307. John Wiley and Sons, 1993.

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

[103] S. Kalenka and N. Jennings. Socially responsible decision making by autonomous agents. In K. Korta, E. Sosa, and X. Arrazola, editors, *Cognition, Agency and Rationality*, pages 135–149. Kluwer Academic Publishers, 1999.

[104] S. Kerr. On the folly of rewarding A, while hoping for B. *Academy of Management Journal*, 18(4):769–782, 1975.

[105] S. Kirn and L. Gasser. Organizational approaches to coordination in multi-agent systems. Technical report, Ilmenau Technical University, Germany, 1998.

[106] S. Kraus, K. Sycara, and A. Evenchik. Reaching agreements through argumentation: A logical model and implementation. *Artificial Intelligence*, 104(1-2):1–69, 1998.

[107] C. Krogh. The rights of agents. In M. Wooldridge, J. P. Müller, and M. Tambe, editors, *Intelligent Agents II (ATAL'95)*, LNAI 1037, pages 1–16. Springer-Verlag, 1996.

[108] N. Lacey and H. Hexmoor. Representing and revising norms and values within an adaptive agent architecture. In *Proceedings of the 2002 International Conference on Artificial Intelligence*. CSREA Press, 2002.

[109] G. Lanzola, S. Falasconi, and M. Stefanelli. Cooperative software agents for patient management. In P. Barahona, M. Stefanelli, and J. Wyatt, editors, *Artificial Intelligence In Medicine (AIME'95)*, LNAI 934, pages 173–184. Springer-Verlag, 1995.

[110] Y. Lashkari, M. Metral, and P. Maes. Collaborative interface agents. In *Proceedings of the Twelfth National Conference on Artificial Intelligence*, volume 1, pages 444–449. AAAI Press, 1994.

[111] V. Lesser and D. Corkill. The distributed vehicle monitoring testbed: A tool for investigating distributed problem solving networks. *AI Magazine*, 4(3):15–33, 1983.

[112] F. López y López and M. Luck. Empowered situations of autonomous agents. In F. Garijo, J. Riquelme, and M. Toro, editors, *Advances in Artificial Intelligence - IBERAMIA 2002. The 8th Iberoamerican Conference on AI*, LNAI 2527, pages 585–595. Springer-Verlag, 2002.

[113] F. López y López and M. Luck. Towards a model of the dynamics of normative multi-agent systems. In G. Lindemann, D. Moldt, M. Paolucci, and B. Yu, editors, *Proceedings of the International Workshop on Regulated Agent-Based Social Systems: Theories and Applications (RASTA'02) at AAMAS'02*, pages 175–193. University of Hamburg, 2002.

[114] F. López y López and M. Luck. Modelling norms for autonomous agents. In *Proceedings of the Fourth Mexican International Conference on Computer Science (ENC'03) (to appear)*. IEEE Computer Society Press, 2003.

[115] F. López y López, M. Luck, and M. d'Inverno. A framework for norm-based inter-agent dependence. In C. Zozaya, M. Mejia, P. Noriega, and A. Sanchez, editors, *Proceedings of the Third Mexican International Conference on Computer Science (ENC'01)*, pages 31–40. SMCC-INEGI, 2001.

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

[117] M. Luck and M. d'Inverno. A formal framework for agency and autonomy. In V. Lesser and L. Gasser, editors, *Proceedings of the First International Conference on Multi-Agent Systems (ICMAS'95)*, pages 254–260. AAAI Press/MIT Press, 1995.

[118] M. Luck and M. d'Inverno. Engagement and cooperation in motivated agent modelling. In C. Zhang and D. Lukose, editors, *Proceedings of the First Australian Workshop on DAI*, LNAI 1087, pages 70–84. Springer-Verlag, 1996.

[119] M. Luck and M. d'Inverno. Motivated behaviour for goal adoption. In C. Zhang and D. Lukose, editors, *Multi-Agents Systems. Theories Languages and Applications*, LNAI 1544, pages 58–73. Springer-Verlag, 1998.

[120] M. Luck and M. d'Inverno. A conceptual framework for agent definition and development. *The Computer Journal*, 44(1):1–20, 2001.

[121] M. Luck and M. d'Inverno. Plan analysis for autonomous sociological agents. In Y. Lesperance and C. Castelfranchi, editors, *Intelligent Agents VII (ATAL'00)*, LNAI 1986, pages 172–186. Springer-Verlag, 2001.

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

[123] T Malone. Modeling coordination in organizations and markets. *Management Science*, 33(10):1317–1332, 1987.

[124] M. Miceli and A. Cesta. Strategic social planning. Looking for willingness in multi-agent domains. In *Proceedings of the 15th Annual Conference of the Cognitive Science Society*, pages 741–746, 1993.

[125] M. Miceli, A. Cesta, and P. Rizzo. Distributed artifical intelligence from a socio-cognitive standpoint: Looking at reasons for interaction. *Artificial Intelligence and Society*, 9:287–320, 1996.

[126] D. Moffat and N. Frijda. Where there's a will there's an agent. In M. Wooldridge and N. Jennings, editors, *Intelligent Agents (ATAL'94)*, LNAI 890, pages 245–260. Springer-Verlag, 1995.

[127] Y. Moses and M. Tennenholtz. Artificial social systems. Technical report CS90-12, Weizmann Institute, Israel, 1990.

[128] H.J. Müller. Negotiation principles. In G. O'Hare and N. R. Jennings, editors, *Foundations of Distributed Artificial Intelligence*, pages 211–229. John Wiley & Sons, 1996.

[129] J. Müller. *The Design of Intelligent Agents: A Layered Approach.* Springer-Verlag, 1996.

[130] J. P. Müller. Architectures and applications of intelligent agents: A survey. *The Knowledge Engineering Review*, 13(4):353–380, 1998.

[131] T. Norman and D. Long. Goal creation in motivated agents. In M. Wooldridge and N. Jennings, editors, *Intelligent Agents (ATAL'94)*, LNAI 890, pages 277–290. Springer-Verlag, 1995.

[132] T. Norman and D. Long. Alarms: An implementation of motivated agency. In M. Wooldridge, J. P. Müller, and M. Tambe, editors, *Intelligent Agents II (ATAL'95)*, LNAI 1037, pages 219–234. Springer-Verlag, 1996.

[133] T. Norman, C. Sierra, and N. Jennings. Rights and commitments in multi-agent agreements. In Y. Demazeau, editor, *Proceedings of the Third International Conference on Multi-Agent Systems (ICMAS'98)*, pages 222–229. IEEE Computer Society Press, 1998.

[134] H. Nwana. Software agents: An overview. *The Knowledge Engineering Review*, 11(3):205–244, 1996.

[135] S. J. Ott. Power and influence. In S. J. Ott, editor, *Classic Readings in Organizational Behavior*, pages 420–428. Brooks/Cole Publishing Company, 1989.

[136] P. Panzarasa, N. Jennings, and T. Norman. Formalising collaborative decision making and practical reasoning in multi-agent systems. *Journal of Logic and Computation*, 12(1):55–117, 2002.

[137] P. Panzarasa, T. Norman, and N. Jennings. Social mental shaping: modelling the impact of sociality on autonomous agents' mental states. *Computational Intelligence*, 17(4):738–782, 2001.

[138] S. Parsons, C. Sierra, and N. Jennings. Agents that reason and negotiate by arguing. *Journal of Logic and Computation*, 8(3):261–292, 1998.

[139] H. Parunak. Manufacturing experience with the contract net. In M. Huhns, editor, *Distributed Artificial Intelligence*, pages 285–310. Morgan Kaufmann, 1987.

[140] M. Pollack. The uses of plans. *Artificial Intelligence*, 57(1):43–68, 1992.

[141] A. Rao and M. Georgeff. Modeling rational agents within a BDI-architecture. In J. Allen, R. Fikes, and E. Sandewall, editors, *Proceedings of the Second International Conference on Principles of Knowledge Representation and Reasoning*, pages 473–484. Morgan Kaufmann Publishers Inc., 1991.

[142] A. S. Rao. AgentSpeak(L): BDI agent speak out in a logical computable language. In W. Van de Velde and J. Perram, editors, *Agents Breaking Away (MAAMAW'96)*, LNAI 1038, pages 42–55. Springer-Verlag, 1996.

[143] E. Rich and K. Knight, editors. *Artificial Intelligence*. Mc Graw-Hill, 1996.

[144] A. Rocha and E. Oliveira. Electronic institutions as a framework for agents' negotiation and mutual commitment. In *Portuguese Conference on Artificial Intelligence*, pages 232–245, 2001.

[145] J. Rosenschein and M. Genesereth. Deals among rational agents. In *Proceedings of the Nineth International Joint Conference on Artificial Intelligence*, pages 91–99. Morgan Kaufmann Publishers, 1985.

[146] A. Ross. *Directives and Norms*. Routledge and Kegan Paul Ltd., 1968.

[147] S. Russell and P. Norvig. *Artificial Intelligence*. Prentice-Hall International Editions, 1995.

[148] N. Saam and A. Harrer. Simulating norms, social inequality, and functional change in artificial societies. *Journal of Artificial Societies and Social Simulation*, 2(1), 1999.

[149] S. Sen. Reciprocity: a fundational principle for promoting cooperative behavior among self-interested agents. In Victor Lesser, editor, *Proceedings on the Second International Conference on Multi-Agent Systems (ICMAS-96)*, pages 322–329. The AAAI Press, 1996.

[150] S. Sen. Believing others: Pros and cons. *Artificial Intelligence*, 142(2):179–203, 2002.

[151] M. Sergot. Normative positions. In P. Mc Namara and H. Prakken, editors, *Norms, Logics and Information Systems*, pages 289–308. IOS Press, 1999.

[152] Y. Shoham. Agent-oriented programming. *Artificial Intelligence*, 60(1):51–92, 1993.

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

[154] J. Sichman, R. Conte, Y. Demazeau, and C. Castelfranchi. A social reasoning mechanism based on dependence networks. In A. Cohen, editor, *Proceedings of the 11th European Conference on Artificial Intelligence (ECAI94)*, pages 188–192. John Wiley & Sons, 1994.

[155] J. Sichman and Y. Demazeau. Exploiting social reasoning to deal with agency level inconsistency. In V. Lesser and L. Gasser, editors, *Proceedings of the First International Conference on Multi-Agent Systems (ICMAS'95)*, pages 352–359. AAAI Press/MIT Press, 1995.

[156] K. Sigmund, E. Fehr, and M. Nowak. The economics of fair play. *Scientific American*, pages 83–87, January 2002.

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

[158] A. Sloman. Motives mechanism and emotions. *Cognition and Emotion*, 1(3):217–234, 1987.

[159] A. Sloman and M. Croucher. Why robots will have emotions. In *International Joint Conference of Artificial Intelligence*, pages 58–73. AAI Press, 1981.

[160] R. Smith. The contract net protocol: High-level communication and control in a distributed problem solver. *IEEE Transactions on Computers*, C-29(12):1104–1113, 1980.

[161] R. Smith and R. Davis. A framework for distributed problem solving. *IEEE Transactions on Systems, Man and Cybernetics*, SMC-11(1):61–70, 1981.

[162] I. Sommerville. *Software Engineering*. Addison-Wesley, 1992.

[163] J. M. Spivey. *The Z Notation: A Reference Manual*. Prentice-Hall, 1992.

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

[165] R. Tuomela and M. Bonnevier-Toumela. Social norms, task, and roles. Technical report HL-97948, University of Helsinki, Helsinki, 1992.

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

[167] E. Ullmann-Margalit. *The Emergence of Norms.* Oxford University Press, 1977.

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

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

[170] A. van Lamsweerde. Formal specification: a roadmap. In A. Finkelstein, editor, *The Future of Software Engineering*, pages 147–159. ACM Press, 2000.

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

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

[173] R. Wieringa and J. Meyer. Applications of deontic logic in computer science: A concise overview. In J. Meyer and R. Wieringa, editors, *Deontic Logic in Computer Science: Normative System Specification*, pages 17–40. John Wiley and Sons, 1993.

[174] T. Wittig, N. Jennings, and E. Mamdani. ARCHON a framework for intelligent cooperation. *IEE-BCS Journal of Intelligent Systems Engineering. Special Issue on Real-time Intelligent Systems in ESPRIT*, 3(3):168–179, 1994.

[175] J. Woodcock and J. Davis. *Using Z : Specification, Refinement, and Proof.* Prentice-Hall, 1994.

[176] M. Wooldridge. Intelligent agents. In G. Weiss, editor, *Multi-Agent Systems: A Modern Approach to Distributed Artificial Intelligence*, pages 27–75. The MIT Press, 1999.

[177] M. Wooldridge and N. Jennings. Intelligent agents: Theory and practice. *The Knowledge Engineering Review*, 10(2):115–152, 1995.

[178] M. Wooldridge and N. Jennings. The cooperative problem solving process. *Journal of Logic & Computation*, 9(4):563–592, 1999.

[179] J.B. Wordsworth. *Software Development with Z.* Addison-Wesley, 1992.