Mechanisms for norm emergence and norm identification in multi-agent societies

Bastin Tony Roy Savarimuthu

a thesis submitted for the degree of
Doctor of Philosophy
at the University of Otago, Dunedin,
New Zealand.

12 August 2011
Abstract

Norms are standards of behaviour expected of the members of a society. Norms in human societies help in sustaining social order and facilitating cooperation and coordination among agents. Researchers in multi-agent systems investigate how the concept of norms can serve a similar purpose in artificial societies with autonomous agents. This thesis contributes to two aspects of the study of norms in multi-agent systems through the investigation of mechanisms for norm emergence and norm identification.

With the advent of digital societies such as Second Life, software agents that reside in these societies are expected to abide by the norms of those societies. Most works on norms in multi-agent systems assume that agents know the norms a priori. Though this is important, norms that are not explicitly specified by the designers of the society may emerge in open agent societies. Thus there is a need for the study of mechanisms for artificial agent societies which can facilitate norm emergence based on interactions between agents. To that end the first part of this thesis describes a role model mechanism for norm emergence. The thesis also describes how norms can emerge in connection with different types of network topologies. A particle-collision model for constructing dynamic network topologies has been applied to model how two societies can be brought together. Using such a model, norm emergence on dynamic network topologies have been studied.

With the uptake of virtual environments which are open and dynamic, agents residing in these societies should be endowed with mechanisms that facilitate norm identification. To that end, the second part of the thesis investigates how a software agent comes to know the norms of the society that it is a part of. To achieve
this goal, the thesis presents an internal agent architecture for norm identification. The architecture provides a mechanism for an agent to infer norms based on observing local interactions and signals (sanctions). The software agents equipped with this architecture will be capable of finding two types of norms, prohibition norms and obligation norms. The thesis demonstrates how an agent in a society is able to add, modify and remove norms dynamically. The thesis also demonstrates that an agent is able to identify conditional norms.

Thus, the contributions of this thesis are to two aspects of the study of norms, norm emergence and norm identification.
Acknowledgements

- First of all, my heartfelt thanks to my supervisors Maryam Purvis, Stephen Cranefield and Martin Purvis for their encouragement and support throughout the eight long years of my PhD which I pursued on a part-time basis.

- I would like to thank my parents Savarimuthu Arockiam and Bilomeenal Amalraj for their persistent enquiry of the status of my PhD and their constant encouragement to move forward towards completion.

- It would have been impossible for me to finish my PhD without the help of my wife Rexy Arulanandam. On many occasions she was happy to care for our son Rohan Savarimuthu, thereby creating slots for me to work on the thesis. My thanks also goes to my sister Sharmila Savarimuthu for proof-reading some chapters of my thesis. I enjoyed discussing and debating several aspects of my thesis with her.

- I would like to thank my friends Mariusz Nowostawski, Antoni Moore and Carl Leichter for their support and suggestions on my PhD.

- Thanks goes to Michael Winikoff for reviewing the background chapters of my thesis. I would also like to thank the members of the New Zealand Distributed Information Systems (NZDIS) group for their feedback on my research work.

- I would also like to thank Christian Müller-Schloer and Peter Whigham for their feedback on the chapter on empirical works on norms. Thanks also goes to the COIN workshop participants and reviewers (2007, 2009 & 2010) and the NorMAS 2009 participants for their valuable feedback. In particular, I would like to thank David Grossi, Harko Verhagen, Pablo Noriega, and Adrian Perreau De Pinninck Bas for the interactions I have had at one of these events.
• I would like to thank James Irwin for proof reading my thesis. I would also like to thank Jeremiah Deng for convening my PhD review meetings.

• My thanks to the fellow travelers of the PhD journey for sharing their experiences of their roller-coaster rides, Toktam Ebadi, Vivian Yong and Prasath Jayakaran.

• I would like to thank all my friends who were a part of this journey at some point or the other, especially Muthu Kumar Thamizh Selvan, Sagheer Ahmed, Poonam Pal, Jaya Pal, Prakash Pal and Marcos De Oliveira.

• My thanks also goes to Heather Harris, Gail Mercer and Brendon Sly for their help in producing hard-copies of this thesis.

• Finally and most importantly, I would like to thank God for making this thesis possible!
## Contents

1 Introduction ................................................................. 1
  1.1 Motivation of the research ........................................ 1
  1.2 Research questions and original contributions ............. 3
    1.2.1 Norm emergence ............................................. 4
    1.2.2 Norm identification ......................................... 4
    1.2.3 Refereed publications ....................................... 5
  1.3 Organization .......................................................... 6

2 Agents and norms ....................................................... 8
  2.1 Software agents and multi-agent systems (MAS) .......... 8
    2.1.1 Software agents ............................................ 8
    2.1.2 Multi-agent Systems (MAS) ............................... 9
    2.1.3 A brief history of multi-agent systems ............... 10
      2.1.3.1 1980s to 2000 ........................................ 11
      2.1.3.2 2000 to 2010 .......................................... 13
    2.1.4 Disciplines influencing multi-agent systems ....... 14
      2.1.4.1 Artificial intelligence ................................ 14
      2.1.4.2 Distributed systems .................................. 15
      2.1.4.3 Economics ............................................... 16
      2.1.4.4 Sociology ............................................... 16
      2.1.4.5 Philosophy ............................................. 16
      2.1.4.6 Software engineering ................................. 17
      2.1.4.7 Perspective of this thesis .......................... 17
  2.2 What are norms? ...................................................... 18
    2.2.1 Norms in human societies ................................ 19
    2.2.2 Conventions vs. social norms vs. laws ............... 21
    2.2.3 Norm life-cycle ............................................ 22
  2.3 Normative multi-agent systems (NorMAS) .................... 23
    2.3.1 Definitions ................................................ 23
    2.3.2 Different views of norms in NorMAS ................... 25
    2.3.3 Two branches of research in NorMAS ................... 27
    2.3.4 Normative architectures .................................. 28
    2.3.5 Norm representation ....................................... 32
    2.3.6 Norm enforcement .......................................... 33
    2.3.7 Discussion .................................................. 33
  2.4 Summary ............................................................... 34
3 Empirical approaches to the study of norms

3.1 Developmental phases of norms based on empirical studies
  3.1.1 Norm creation
  3.1.2 Norm identification
  3.1.3 Norm spreading
  3.1.4 Norm enforcement
  3.1.5 Norm emergence
  3.1.6 Consideration of network topologies
  3.1.7 Discussion

3.2 Categorization of empirical works on norms
  3.2.1 Off-line design approaches
  3.2.2 Leadership and entrepreneurship mechanisms
  3.2.3 Learning mechanisms
    3.2.3.1 Imitation mechanisms
    3.2.3.2 Works based on machine learning
    3.2.3.3 Data mining mechanism
    3.2.3.4 Discussion
  3.2.4 Cognitive approach
  3.2.5 Cultural and evolutionary mechanisms
  3.2.6 Sanction mechanisms
  3.2.7 Reputation mechanisms
  3.2.8 Emotion-based mechanisms
  3.2.9 Research using network topologies
    3.2.9.1 Research using a static network topology
    3.2.9.2 Dynamic topology works
    3.2.9.3 Discussion

3.3 Characteristics of empirical works on norms
  3.3.1 Characteristics of an agent
    3.3.1.1 Norm identification
    3.3.1.2 Norm internalization
    3.3.1.3 Norm violation
    3.3.1.4 Norm influence
    3.3.1.5 Norm representation
    3.3.1.6 Norm communication
    3.3.1.7 Interaction context
  3.3.2 Characteristics of an agent society
    3.3.2.1 Nature of the society and its size
    3.3.2.2 Consideration of network topology
    3.3.2.3 Consideration of norm coexistence
    3.3.2.4 Consideration of norm change
    3.3.2.5 Consideration of norm conflicts
    3.3.2.6 Norm emergence criteria
    3.3.2.7 Other characteristics
  3.3.3 Discussion

3.4 Contextualizing the contributions of this thesis
  3.4.1 Norm emergence
4 Role model based mechanism for norm emergence

4.1 Introduction .................................................. 80
4.2 Role model mechanism ...................................... 81
4.3 Social network topologies .................................... 84
  4.3.1 Fully connected network : ............................... 84
  4.3.2 Random network : ....................................... 85
  4.3.3 Scale-free network : ..................................... 85
  4.3.4 Some characteristics of networks : .................... 86
4.4 Interaction context and the experimental set-up ............ 87
  4.4.1 Interaction context ...................................... 87
    4.4.1.1 The Ultimatum game .............................. 88
  4.4.2 Description of the multi-agent environment ............ 88
    4.4.2.1 Norms in the society ............................. 88
    4.4.2.2 Autonomy of agents ............................... 89
    4.4.2.3 Interaction between agents ....................... 90
4.5 Experiments and results .................................... 92
  4.5.1 Demonstration of the emergence of leadership networks and norms 92
  4.5.2 Norm emergence on top of random and scale-free networks .......... 94
  4.5.3 Power law behaviour of the leadership network ........... 97
4.6 Discussion .................................................. 97
4.7 Summary ................................................... 98

5 Norm emergence in dynamically changing networks

5.1 Introduction .................................................. 100
5.2 Architecture of the experimental setup for norm emergence ...... 101
  5.2.1 Mobile agent model ..................................... 102
5.3 Norm emergence in a single agent society ..................... 103
5.4 Extending mobile agent model for bringing two societies together 105
5.5 Norm emergence when two societies are brought together .......... 106
5.6 Discussion .................................................. 110
5.7 Summary ................................................... 111

6 Architecture for norm identification

6.1 Introduction .................................................. 112
6.2 The need for a norm identification framework .................. 112
  6.2.1 Scenario 1 - Virtual environments such as Second Life .......... 113
  6.2.2 Scenario 2 - e-commerce environments ........................ 113
  6.2.3 Scenario 3 - Massively multi-player online games ............... 113
6.3 The process of norm identification ................................ 114
6.4 Overview of the components of the framework ................... 118
List of Tables

3.1 Mechanisms used in different phases of the norm life-cycle .......................... 42
3.2 Comparison of characteristics of an agent in the simulation works on norms .... 64
3.3 Comparison of characteristics of an agent society in the simulation works on norms .......................... 70

7.1 Events observed by an agent ........................................................................... 128
7.2 Candidate prohibitions norms on varying Norm Identification Threshold ... 133

8.1 Simulation parameters for identifying the obligation norm of tipping ................ 168

9.1 Simulation parameters for identifying conditional norms ................................ 187

B.1 Acronyms used and the corresponding expansions .......................................... 240
## List of Figures

2.1 A three level view of multi-agent systems .......................... 11  
2.2 Interdisciplinary nature of multi-agent systems .................. 15  
2.3 Fields of study of norms .............................................. 18  
2.4 Progression from conventions to laws ................................. 22  

3.1 Three main stages of the norm life-cycle .......................... 37  
3.2 Phases of norm life-cycle and categories of empirical models ........ 38  
3.3 Categorization of empirical works on norms ........................ 45  
3.4 Higher level overview of interactions between agents ............... 75  
3.5 Contributions of the thesis to different categories of empirical works on norms 77  

4.1 Two layers of networks used in role model agent mechanism ........ 84  
4.2 A fully connected network and a random network ................. 85  
4.3 An Barabasi-Albert scale-free network with 50 nodes .............. 86  
4.4 Two layers of networks used in role model agent mechanism ......... 91  
4.5 Snapshots of the emergence of leadership and norms in an agent society 93  
4.6 Norm convergence in ER networks when the average degree of connectivity is varied ......................................................... 94  
4.7 Comparison of norm convergence in random vs scale-free networks .... 95  
4.8 Power law behaviour of the leadership network ..................... 96  

5.1 Architecture of the experimental setup for norm emergence ........ 101  
5.2 A dynamically constructed network with 100 mobile agents .......... 103  
5.3 Norm emergence on varying the relationship duration .............. 104  
5.4 A dynamically constructed network with 50 mobile agents in two groups before and after the collapse of the wall between them .......... 105  
5.5 Graph showing the norm convergence results when two societies are brought together (TTL = 100) ............................................. 107  
5.6 Graph showing the convergence results when two societies are brought together (TTL = 50) .............................................. 108  
5.7 Graph showing the norm convergence results when two societies of different sizes are brought together (TTL = 100) ......................... 109  

6.1 Architecture of the norm identification framework of an agent ........ 115  
6.2 Overview of the norm inference component ........................... 122  

7.1 A sample special event episode list (SEEL) .......................... 129  
7.2 Two sets of norms .......................................................... 136
A.1  Graph showing the norm convergence results when two societies of different sizes are brought together (TTL = 200) . . . . . . . . . . . . . . . . . . . . . . . . . . . . 237
A.2  Graph showing the norm convergence results when two societies of different sizes are brought together (TTL = 50) . . . . . . . . . . . . . . . . . . . . . . . . . . . . 238
List of Algorithms

4.1 Pseudocode of an agent to find a normative advisor ........................................ 83
7.1 Overview of the norm inference process .............................................................. 127
7.2 Pseudocode to create Event Episode List ......................................................... 129
7.3 Pseudocode to create Candidate Prohibition Norms Set (CPNS) .................... 131
7.4 Pseudocode to create Sub-Episode List ............................................................. 132
8.1 Obligation Norm Inference algorithm (main algorithm) ....................................... 159
8.2 Pseudocode to create the Candidate Obligation Norm Set (CONS) ................. 163
9.1 Pseudocode of an agent to identify conditional norm associated with the pro-
hibition of littering ......................................................................................... 182
9.2 Pseudocode of an agent to identify a conditional norm ..................................... 184
Chapter 1

Introduction

Norms have played an important role in regulating the behaviour of individuals in human societies. For example, regulation of behaviour in public spaces such as restaurants is governed by norms. It is expected that no one smokes in a restaurant and one might be obliged to tip a waiter for good service. Thus norms are guidelines for actions that apply to specific situations (e.g. prescription and proscription of certain behaviour in a social setting). Norms also facilitate cooperation (Elster, 1989b) and coordination (Bicchieri, 2006) in human societies. Norms are fundamental to the understanding of social control and its variation across societies (Campbell, 1964). Hence it is not surprising that norms are of interest to researchers in different disciplines such as sociology, economics, anthropology and computer science.

1.1 Motivation of the research

Inspired by human societies, researchers in multi-agent systems have investigated how the concept of norms can be used to constrain the behaviour of software agents (López y López, 2003) and also improve cooperation and coordination among agents (Shoham and Tennenholtz, 1992). Norms are simple constructs that have also been shown to reduce the amount of computation required by software agents, since on identifying norms agents can follow them without much “thought” (Epstein, 2001).

Researchers in multi-agent systems are interested in norms from two perspectives, the top-down and bottom-up approaches. In the top-down approach to norms, researchers have
investigated how normative architectures and systems can be constructed for improved social behaviour. Constraining the actions that agents can perform can promote a smoother functioning society (e.g. constraining agents from littering a park). In the bottom-up approach to norms, researchers are interested in studying how norms may spread and emerge in agent societies based on their interactions with other agents in the society.

With the advent of the Internet, software agents typically exist in electronic societies that are open and dynamic. Currently, multi-agent system researchers investigate how norm-based mechanisms can be used to facilitate social control in these electronic societies. Examples of norm-governed electronic societies include electronic institutions (Arcos et al., 2005a), virtual societies such as Second Life (Cranefield and Li, 2009) and massively multiplayer online games (MMOGs) (Johansson and Verhagen, 2009). Researchers are also investigating mechanisms for norm emergence in these societies (Sen and Airiau, 2007; Salazar et al., 2008).

One of the limitations with the existing works on the top-down perspective of norms is that most works have concentrated only on how norms regulate behaviour (López y López, 2003; Gaertner et al., 2007). These works assume that agents somehow know (a priori) what the norms of a society are (e.g. through off-line design). For example, an agent may have obtained the norm from a leader (Boman, 1999) or through an institution that prescribes what the norms of the society should be (Shoham and Tennenholtz, 1995; Conte and Castelfranchi, 1995; Aldewereld et al., 2006a; García-Camino et al., 2006b; Vázquez-Salceda, 2003). However, it may not be possible for an agent designer to specify all possible norms that agents may encounter in an open agent society. As the composition of the open agent society changes, the norms may change. Agents in electronic societies can join and leave different societies where the norms may be diverse. For these reasons, mechanisms for norm emergence should be investigated. To this end the first part of the thesis aims at studying a role model mechanism for norm emergence. The role model mechanism focuses on how an agent society can adopt a norm using a distributed leadership mechanism. Additionally, the relationships between agents in a society can change dynamically (e.g. based on agents moving in and out of societies). In this thesis the relationships between agents in a society are viewed in terms of a network topology, with nodes representing agents and links representing relationships. The role model mechanism is applied with re-
spect to both static and dynamic network topologies to investigate how norms emerge in these networked societies.

Another limitation of most works on norms is the lack of an explicit norm identification mechanism for an individual agent to recognize or identify norms. Since norms in an open agent society might change dynamically, an agent should have the capability to recognize a norm when it is established and also recognize when it changes. The agents should also be able to add, modify, and remove norms. **To that end the second part of this thesis discusses the internal agent architecture for norm identification.** An agent using this architecture uses an association mining approach to identify two types of norms, prohibition norms and obligation norms. Prohibition norms and obligation norms are identified using Prohibition Norm Identification (PNI) and Obligation Norm Identification (ONI) algorithms, respectively. The thesis also describes how conditional norms can be identified in an agent society.

### 1.2 Research questions and original contributions

The thesis contributes to two aspects of normative multi-agent systems research, *norm emergence* and *norm identification*. These two main contributions presented in the form of research questions are given below. The sub-parts to these questions are also described.

1. How can we design and develop a mechanism for norm emergence that employs a role model mechanism (a distributed leadership approach)?

   (a) What is the effect of static network topologies on norm emergence, where the interaction between agents are based on the role model mechanism?

   (b) How can we model the creation of dynamic network topologies? How can the bringing together of two societies be modeled, and what is the impact of applying the role model mechanism on societies that are evolving (i.e. dynamically changing societies)?

2. How can we design an architecture for an agent to infer norms in an open agent society? Using the designed architecture, can we answer the following questions?
(a) How can an agent infer prohibition norms?

(b) How can an agent infer obligation norms?

(c) How can an agent identify conditional norms?

The research questions presented above are elaborated in sub-sections 1.2.1 and 1.2.2.

1.2.1 Norm emergence

Works on norm emergence that are based on leadership have used a centralized approach (Boman, 1999; Verhagen, 2000) where an agent asks for normative advice from a normative advisor of the society. This thesis presents a role model mechanism that is based on a distributed leadership approach which emulates a real-life society having a hierarchy of leaders and followers: a leader follows another leader who in turn can follow some other leader. The role of static network topologies on norm emergence is also investigated, where the interactions between agents are based on the role model mechanism. The thesis also describes the use of a particle-collision model to model mechanically the dynamic evolution of an agent network topology and also to model what happens when two agent societies come into contact. Additionally, the role model mechanism is also tested on top of dynamic network topologies.

It should be noted that the work on norm emergence is from an external viewpoint of the society (i.e. a bird’s eye view on the emergence of norms in the society) where an external agent observes the creation, spread, and the emergence of norms in an agent society.

1.2.2 Norm identification

Some works on norms have assumed that agents know the norms of the society (Conte and Castelfranchi, 1995). Most game-theory based works consider a norm in terms of a utility maximizing strategy (Shoham and Tennenholtz, 1992; Sen and Airiau, 2007). Not many researchers have considered the question of how an agent can explicitly identify norms in an open agent society. To that end, the second part of this thesis examines how an agent can infer norms. Thus an internal agent architecture for norm identification has been proposed and studied here. This architecture enables an agent to identify two types of norms, prohibition
norms and obligation norms. The details on how an agent identifies a prohibition norm are explained in the context of a public park scenario, where the norm against littering is identified by the agent. The obligation norm inference is explained in the context of a tipping norm in a restaurant scenario. Experimental results on the identification of these two types of norms are discussed. We also demonstrate that an agent using the proposed architecture can dynamically add, remove and modify norms based on observing the environment. The agents can identify co-existing and conditional norms. The agents can also identify the normative pre-conditions (conditions that have to be true for the norm to hold). We also demonstrate that the utility of an agent is better if it stays in a society where norms are enforced (i.e. punishers are present).

It should be noted that the work on norm identification is from the viewpoint of a single agent based on the agent’s observation of the interactions between agents in its local environment.

1.2.3 Refereed publications

The following publications have resulted from various chapters of this thesis.

Journal articles


The thesis is organized as follows. Chapter 2 provides a background on norms and how the concept of norms is investigated in the field of normative multi-agent systems (NorMAS). Chapter 3 presents a life-cycle model based on the developmental phases of norms and also provides an overview of categories of mechanisms employed by empirical works on norms. It also describes the categories to which this thesis contributes.

The two focus areas of this thesis are norm emergence and norm identification. The work on norm emergence is reported in Chapters 4 and 5. Chapter 4 describes the role model mechanism for norm emergence. Chapter 5 describes how dynamic network topologies are constructed and how norms emerge on top of dynamic network topologies. This chapter also describes how two societies can be brought together and explains the norm dynamics in the newly formed society.
Our contributions to the area of norm identification are reported in Chapters 6 to 9. Chapter 6 provides an overview of the internal agent architecture for norm identification. Chapters 7 and 8 describe the process of identifying prohibition and obligation norms respectively. Chapter 9 describes how conditional norms are identified.

Chapter 10 provides a discussion on the contributions of this thesis, its limitations, and the future work. Concluding remarks are presented in Chapter 11.
Chapter 2

Agents and norms

The objective of this chapter is to provide an introduction to software agents and how the concept of norms is being studied by multi-agent systems (MAS) researchers. An overview of the field of multi-agent systems (MAS) is provided in Section 2.1. A background on the concept of norms in human societies is provided in Section 2.2. An introduction to the field of normative multi-agent systems (NorMAS) including normative architectures, norm representation and norm enforcement is provided in Section 2.3.

2.1 Software agents and multi-agent systems (MAS)

This section provides an overview of the field of multi-agent systems. Sections 2.1.1 and 2.1.2 provide an introduction to software agents and multi-agent systems respectively. Section 2.1.3 provides a brief history of multi-agent systems. Section 2.1.4 discusses the interdisciplinary nature of the research carried out in multi-agent systems.

2.1.1 Software agents

The concept of software agents was first conceived by John McCarthy in mid-1950s, and the term software agents was later coined by Oliver G. Selfridge (Kay, 1984). A software agent in their view was a system that carried out some operations to achieve a given goal and can ask for and receive advice when required from humans or other agents.

Several definitions for software agents exist (Nwana, 1996; Wooldridge and Jennings,
1995; Shoham, 1997; Franklin and Graesser, 1997). A widely accepted definition by Michael Wooldridge (2009) is that an agent is a computer system that is situated in some environment, and that is capable of autonomous action in this environment in order to meet its delegated objectives.

Several characteristics of software agents are defined by researchers. Software agents are autonomous (i.e. have the ability to decide whether to pursue an action without human intervention), reactive (i.e. are able to perceive the environment and take appropriate action) pro-active (i.e. have the ability to take initiative in order to meet an objective) and have social ability (i.e. the capability of communicating with other agents) (Wooldridge, 2009). Other attributes of agents include the abilities to cooperate, collaborate, negotiate, learn and adapt. Agents in a society may possess different personalities. Some agents may have the ability to migrate from one place to another (mobile agents).

Software agents simplify the complexity of distributed computing (Bradshaw, 1997). This is achieved by agents playing different roles in performing different activities. Agents can act as resource managers (e.g. managing databases and web services (Huhns, 2002)). There are different kinds of agents employed by researchers in multi-agent systems. The agents include the user-interface agents that perform certain tasks on behalf of the user, such as finding the best price on items (Doorenbos et al., 1997; Chavez and Maes, 1996), trading agents (Stone et al., 2001; MacKie-Mason and Wellman, 2006), learning agents (Weiss and Sen, 1996; Panait and Luke, 2005), virtual agents (Belliika et al., 1998; Gratch et al., 2007) and monitoring and controlling agents. There are several types of monitoring agents employed by researchers, such as the vehicle monitoring agents in space (Durfee, 1988), governor agents that monitor messages (Arcos et al., 2005b), and agents that monitor workflows (Huhns and Singh, 2002; Wang and Wang, 2006).

2.1.2 Multi-agent Systems (MAS)

A multi-agent system (MAS) is a distributed system that is composed of different types of agents with various capabilities that interact with each other by sending and/or receiving messages (Wooldridge, 2009). Wooldridge notes that the agents in a multi-agent system may represent or may act on behalf of users or owners with very different goals and motivations.
Additionally, in order to successfully interact, these agents will require the ability to cooperate, coordinate, and negotiate with each other. An example of the multi-agent system is a team of robots with software agents embedded in them which can be used for rescue operations (Kitano et al., 1999). The agents in this scenario will have to communicate, cooperate, coordinate, and negotiate with each other.

An agent in a multi-agent system typically operates in an environment where it only has a partial view of the world (i.e. partial information is available). Additionally, the actions it performs in the environment may be non-deterministic (i.e. outcomes are unpredictable). The agent operates in a dynamic environment and the environment can have an infinite state space. Environments that have these four attributes are considered to be open environments (Hewitt, 1986).

A multi-agent system can be viewed as being composed of three levels, as shown in Figure 2.1: the agent level, the interaction level and the society level (Luck et al., 2005). Individual agents situated in an environment have certain capabilities (e.g. the abilities to: observe the world, reason about the world, learn based on observation and plan for actions to be performed). While working on a complex task, a single agent may not be able to accomplish a given task all by itself. It may need the cooperation of other agents, which requires interactions between agents. An agent interacts with other agents by communicating with them (e.g. by sending a request for some information). The interactions between agents may be governed by protocols such as communication protocols (e.g. interaction protocols specified by FIPA (2002)) and negotiation protocols (Kraus, 1997; Faratin et al., 1998; Jennings et al., 2001). Interactions between agents take place within an underlying structure such as a social or an organizational structure. Concepts such as norms, trust, and reputation operate at this higher level, the society level. These three levels can be reduced to two levels by considering agents and interactions together at one level (the micro level) and the society at the second level (the macro level).

### 2.1.3 A brief history of multi-agent systems

This section provides a brief history of multi-agent systems. A summary of the history of the field from its beginnings to year 2000 is provided based on the view of the field by
experts (Sycara, 1998; Weiss, 2000; Wooldridge, 2009) in Section 2.1.3.1. Our view of the developments in the field since 2000 is discussed in Section 2.1.3.2.

2.1.3.1 1980s to 2000

The origin of the field of multi-agent systems can be traced to the emergence of a new community called Distributed Artificial Intelligence (DAI) in the 1980s (Davis, 1980). The objective of DAI was to make use of the distributed processors to solve a given problem. Several workshops were organized and the first books on this field were published by this community. In the early 1980s, the research works in DAI assumed that the distributed entities (or agents) shared a common goal. In the mid-1980s it was realized that this would be an ideal assumption for a society of benevolent agents. However, autonomous agents may be self-interested (self-interested agents are similar to the rational agents as studied in the field of economics) and may not be interested to pursue actions that contribute towards a common goal (Rosenschein and Genesereth, 1985). In the later years, this led to the study of mechanisms that could facilitate cooperation (Doran et al., 1997) and collaboration (Wong et al., 1997) among self-interested agents.
In the late 1980s and early 1990s one of the focus areas of research in AI was on building expert systems. In expert systems, expert agents contribute towards solving problems and also offered advice in their domains of speciality. In traditional expert systems, the expert agents lacked the social ability to cooperate and coordinate. DAI researchers contributed towards making expert agents from different expert systems collaborate (Wittig, 1992).

The uptake of Internet in the early 1990s demonstrated that various distributed systems across the globe can be connected with ease. Researchers in multi-agent systems realized the potential for building front-end applications such as shopping assistants that can extract and operate on data from the Internet on behalf of the users. The advent of Internet thus led to the development of the area of agent-mediated electronic commerce, where agents negotiate with each other and participate in auctions on behalf of the users. In the context of agents talking to each other, researchers realized the need for creating a standard for agents to communicate and share knowledge with each other. To this end, the mid-1990s saw an effort towards the standardization of knowledge-sharing mechanisms employed by agents. This led to the development of two agent communication languages: Knowledge Query and Manipulation Language (KQML) and Knowledge Interchange Format (KIF) (Patil et al., 1992). In an effort to provide a formal standardization of agent communication languages, the Foundation for Intelligent Physical Agents (FIPA) was established in 1996.

As well as standards being developed, the mid-1990s saw some researchers investigating how multi-agent systems can be used to model human societies to gain some insights from agent-based simulations (Conte and Gilbert, 1995). Since then researchers have been investigating how the constructs found in human societies, such as trust, reputation, and norms, can be better understood through simulations and also how these can be used to model artificial agent societies (Ramchurn et al., 2004; Mui et al., 2002; Shoham and Tennenholtz, 1992; Walker and Wooldridge, 1995).

In the late 1990s, researchers applied agent technology to complex real-world domains such as robotics, where robots were controlled by embedded software agents. An outcome of the efforts in this area is the establishment of the RoboCup soccer tournament (RoboCup, 2010). The tournament serves as a test-bed for assessing the performances of teams of robots that play soccer. The area of robotics remains one of the active research areas in the field of multi-agent systems.
## 2.1.3.2 2000 to 2010

In the late 1990s and in early 2000s, there was an interest in the standardization of concepts in the field of multi-agent systems. Several monographs (Bradshaw, 1997; Wooldridge, 2003) and edited collections of articles on agents (Huhns and Singh, 1998; White, 1998; Weiss, 2000) were published. This period also saw the growth of multi-agent systems in several different fields, such as mechanism design, electronic institutions, software engineering methodologies, and development of agent frameworks. Wooldridge (2009) notes that since 2001 the field of multi-agent systems has taken off as a mainstream area of computer science.

Also in the early 2000s, with the advent of virtual environments such as Second Life (Rymaszewski et al., 2006), there has been an interest among researchers in the new area of virtual agents, which adds a human-like face to software agents\(^1\). Researchers are interested in modelling a software agent that is capable of displaying emotions, gestures and postures that are exhibited by humans (Burgoon et al., 2000; de Diesbach and Midgley, 2007). Researchers have used psychological theories to develop virtual agents that display human-like behaviour. One such example is the emotion and adaptation model based on appraisal theory (Lazarus, 1991) proposed by Marsella and Gratch (2009). Virtual environments allow for modelling and simulating virtual human societies and also various forms of interactions between agents (i.e. through emotions, gestures, postures, speech and text). Additionally, humans and virtual agents can interact in this new environment, which creates interesting opportunities for researchers to study mechanisms such as trust, reputation, and norms in these environments (e.g. learning norms from each other).

In the recent years, the number of active sub-research areas in the field of multi-agent

\(^{1}\text{Many research works have shown that humans in general are more inclined to interact with anthropomorphic entities as opposed to limited media such as text or voice-only media (Walker et al., 1994; Kramer et al., 2003; Shneiderman and Plaisant, 2005). However, some research have shown that there is no significant difference in performance between users who interacted with anthropomorphic entities and those who interacted with a text-based system (Dehn and van Mulken, 2000). Some of the advantages include the programs being more attractive and enjoyable to normal users (Shneiderman and Plaisant, 2005) and the ability of anthropomorphic entities to retain the attention of the user (Walker et al., 1994). Some of the disadvantages include confusing users on who is responsible for a system’s actions (Shneiderman and Plaisant, 2005) and distracting them from performing tasks (Takeuchi and Naito, 1995).}
systems has grown. Sub communities that study different aspects of multi-agent systems have emerged (e.g. the NorMAS community that studies norms (Boella et al., 2006, 2008), agents and the data-mining community that is interested in the application of data-mining for agents (Cao et al., 2007)). The burgeoning growth is evident from the number of specialized workshops organized in the premier AAMAS conference (a count of 30 in AAMAS 2010 (Workshops at AAMAS, 2010)).

2.1.4 Disciplines influencing multi-agent systems

As indicated in the previous section (Section 2.1.3), the field of multi-agent systems has been shaped by the cross fertilization of ideas from different disciplines. Various concepts, theories, and practices from other disciplines such as artificial intelligence, distributed systems, economics, philosophy and social science have influenced the field of multi-agent systems (shown in Figure 2.2). At the same time the field of multi-agent systems has also contributed towards the advancement of these fields. The strength of multi-agent systems lies in integrating certain aspects of these disciplines that have not been considered before. Integration of different ideas from disparate disciplines in multi-agent systems has enabled successful modeling of real world entities that have individual agents which interact in an organizational or social setting. Researchers realized that the complexity of developing software systems can be reduced by using the multi-agent systems concepts, which has led to the development of a new paradigm for designing and developing multi-agent systems, namely the agent-oriented software development (Shoham, 1997; Jennings and Wooldridge, 2000).

2.1.4.1 Artificial intelligence

Although the beginnings of the multi-agent systems field can be traced to the AI discipline in general (DAI in particular), the focus of AI has traditionally been in the development of intelligence in an individual agent. Planning, learning etc. were studied in sub-communities as independent problems. Many real life scenarios need agents to possess all these capabilities at the same time. In MAS, researchers questioned whether the view intelligence alone is adequate (i.e. without considering the social aspects) for modelling agents. They realized that apart from the individual intelligence of humans, their collective social-ability to
Interdisciplinary nature of multi-agent systems has enabled them to operate effectively in societies. MAS researchers then began investigating the integration of social-aspects associated with software agents.

### 2.1.4.2 Distributed systems

Multi-agent systems are a subclass of concurrent (distributed) systems, but are distinct from the traditional view of distributed systems in that they are autonomous (ability to make independent decisions that satisfy their design objectives) and they may be self-interested agents. In traditional distributed systems, all the distributed components are expected to behave correctly according to a common plan (i.e. they are not autonomous) and they share a common goal (i.e. each component is not interested in maximizing its welfare). However, an agent in charge of a distributed node (such as a web service) may not necessarily share the common goal. Such an agent may want to maximize its own utility and hence may not honour a request for cooperation. Thus, in MAS, these distributed entities are viewed as autonomous, goal-oriented agents.
2.1.4.3 Economics

The perspective of economists is that individual agents are rational (i.e. utility maximizers). The field heavily uses game theory concepts to arrive at optimal solutions (e.g. Nash equilibrium). However, they ignore that in real life humans may not be utility maximizing at all times (e.g. consider the establishment of fair-splits of money between agents when playing Ultimatum games (Nowak et al., 2000)). Additionally, when many agents are involved, arriving at an optimal solution may be difficult. This can be modelled using agent-based simulations (Sichman et al., 1998) involving many agents as studied in the field of agent based computational economics (Tesfatsion, 2006).

2.1.4.4 Sociology

The study of human societies has inspired the modelling of artificial societies. Artificial agent societies composed of software agents can make use of concepts found in human societies such as cooperation, coordination, and negotiation. Social concepts and mechanisms such as trust, reputation, norms, commitments, voting, gossip and ostracism are used to model and study both human and artificial agent societies (Bonabeau, 2002). Sociologists use agent-based tools to build experimental models and study the effect of these models through simulations, while the MAS researchers use these concepts to build operational multi-agent societies (e.g. electronic institutions) using these concepts.

2.1.4.5 Philosophy

Several philosophical ideas have been integrated into the multi-agent system research. For example the theory of speech acts as studied in the philosophy of language has been used to model agent communication. Speech act theory views utterances as actions and considers the way in which agents use the meaning of the utterance to base their actions. Another example is the intentional notion of an agent system. An intentional system is a system whose behaviour can be predicted by the method of attributing belief, desires and rationality (Dennett, 1989). Successful agent models such as Belief-Desire-Intention (BDI) are loosely based on the theory of practical reasoning of human agents developed by philosopher Michael Bratman (1987). Researchers have used different types of logic to formulate
their philosophical stances of agents (BDI logics by Rao (1996) and deontic logic, the logic of permissions, prohibitions and obligations by von Wright (1951) as studied in the field of Normative multi-agent systems (van der Torre, 2003; Boella et al., 2006)).

2.1.4.6 Software engineering

The field of MAS has led to the development of new ways of designing and developing systems that operate in open, dynamic and complex environments. These environments require agents that can reduce complexity by carrying out tasks that are distributed, in their own autonomous fashion. Though object-oriented systems make use of the message passing metaphor, this is restricted to explicit invocation of function calls, where an object does not have the ability to say no to any invocations. Agent interactions are designed using a more authentic message passing metaphor, where a received message is interpreted by an agent based on an agreed upon vocabulary (e.g. ontology). The agent can then make decisions based on its desires, intentions and goals. The agent technology has also stimulated several software engineering efforts such as the development of methodologies for designing agent systems, building industrial-scale agent frameworks and the testing and verification of developed systems. Some of the well-known agent-based software development methodologies include Gaia (Wooldridge et al., 2000), Tropos (Bresciani et al., 2004) and Prometheus (Padgham and Winikoff, 2003) (see (Dam and Winikoff, 2004) and (Sturm and Shehory, 2004) for a comparison of methodologies). Examples of industrial scale agent frameworks are JACK (Busetta et al., 1999) and Jade (Bellifemine et al., 1999). Verification of agent systems using formal methods and testing of developed systems are investigated by some researchers (Coelho et al., 2007; Winikoff and Cranefield, 2010; Winikoff, 2010).

2.1.4.7 Perspective of this thesis

The focus of this thesis is on norms, particularly social norms that apply to a society of software agents. The concept of social norms have been shown to facilitate cooperation (Elster, 1989b; Axelrod, 1986, 1997; Shoham and Tennenholtz, 1992) and coordination (Lewis, 1969; Bicchieri, 2006) between agents in both human and software agent societies. Inspired by the works on norms in various disciplines (mainly sociology and philosophy), normative
multi-agent system researchers (Boella et al., 2006, 2008) investigate norm-based mechanisms, technologies and tools for social control in software agents. Since the focus of this thesis is on the study of certain aspects of norms (norm emergence and norm identification) in software agent societies, the reminder of this chapter will provide a background on norms both in human societies (Section 2.2) and multi-agent societies (Section 2.3).

### 2.2 What are norms?

Norms are expectations of an agent about the behaviour of other agents in the society (Bicchieri, 2006). Human society follows norms, such as the exchange of gifts at Christmas. Norms have been so much a part of different cultures, it is not surprising that it is an active area of research in a variety of fields (see Figure 2.3) including Sociology (Elster, 1989a; Coleman, 1990; Hechter and Opp, 2001), Economics (Akerlof, 1980; North, 1990; Epstein, 2001), Biology (Axelrod, 1986; Boyd and Richerson, 2002; Nakamaru and Levin, 2004; Chalub et al., 2006), Philosophy (von Wright, 1963; Habermas, 1996; Bicchieri, 2006), Law (Ellickson, 1991, 1998) and Computer Science (Shoham and Tennenholtz, 1992; Walker and Wooldridge, 1995).

Figure 2.3: Fields of study of norms


2.2.1 Norms in human societies

Due to the multi-disciplinary nature of norms, several definitions for norms exist. Habermas (1985), a renowned philosopher, identified norm-regulated actions as one of the four action patterns in human behaviour. A norm to him means fulfilling a generalised expectation of behaviour, which is a widely accepted definition for social norms. A behavioural expectation is generalized if every member of a social group expects all others to behave in a certain way in a given situation. Ullmann-Margalit (1977) describes a social norm as a prescribed guide for conduct or action which is generally complied with by the members of the society. She states that norms are the result of complex patterns of behaviour of a large number of people over a protracted period of time. Coleman (1990) writes “I will say that a norm concerning a specific action exists when the socially defined right to control the action is held not by the actor but by others”. Elster notes the following about social norms (Elster, 1989b): “For norms to be social, they must be shared by other people and partly sustained by their approval and disapproval. They are sustained by the feelings of embarrassment, anxiety, guilt and shame that a person suffers at the prospect of violating them. A person obeying a norm may also be propelled by positive emotions like anger and indignation ... social norms have a grip on the mind that is due to the strong emotions they can trigger”. What is common in these definitions is the expectation that an agent behaves in a certain way in a situation and the appropriate behaviour is dictated by the group. There is not any consensus on the level of social control on norm violation. Habermas’s definition does not talk about norm violation. Coleman’s definition mentions social control without any specifics. Elster’s definition explicitly mentions the approval and disapproval of agents on other agent’s behaviour.

Researchers have divided norms into different categories. Tuomela (1995) has grouped norms into two categories: social norms and personal norms. Social norms define the behaviour of the group and are associated with sanctions. Personal norms are based on the personal beliefs of the individuals. Personal norms are the potential social norms. These norms could become social norms if they were to be observed by other agents and if sanctions were associated with not following the norm. Social norms are further classified into r-norms (rule norms) and s-norms (social norms). Personal norms are categorised into m-
norms (moral norms) and p-norms (prudential norms). Rule norms are imposed by an authority based on an agreement between the members (e.g. one has to pay taxes). Social norms apply to large groups such as a whole society and they are based on mutual belief (e.g. one should not litter). Members of a society expect that a social norm be followed by other members of the society. Moral norms appeal to one’s conscience (e.g. one should not steal or accept bribes). Prudential norms are based on rationality (e.g. one ought to maximize one’s expected utility). When members of a society violate societal norms, they may be punished or even ostracised in some cases (de Pinninck et al., 2008).

Many social scientists have studied why norms are followed. Some of the reasons for norm adherence include:

- fear of authority or power (Axelrod, 1986)
- rational appeal of the norms (Akerlof, 1976; Becker, 1978)
- emotions such as shame, guilt and embarrassment that arise because of non-adherence (Elster, 1989b)
- willingness to follow the crowd (Epstein, 2001).

In this thesis, we focus on social norms because the agents in multi-agent systems have been modelled using ideas borrowed from social concepts such as speech act theory (Searle, 1969), collaboration and cooperation (Nwana, 1996).

Based on the definitions provided by various researchers, we note that the notion of a social norm is generally made up of the following three aspects:

- **Normative expectation of a behavioural regularity**: There is a general agreement within the society that a behaviour is expected on the part of an agent (or actor) by others in a society, in a given circumstance.

- **Norm enforcement mechanism**: When an agent does not follow the norm, it could be subjected to a sanction. The sanction could include monetary or physical punishment in the real world which can trigger emotions (embarrassment, guilt, etc.) or direct loss of utility. Other kind of sanctions could include agents not being willing to interact
with an agent that violated the norm or the decrease of its reputation score. Agents that follow the norm might be rewarded.

- **Norm spreading mechanism**: Examples of norm spreading factors include the advice from powerful leaders and entrepreneurs, and the cultural and evolutionary influences. For an external observer, agents identifying and adopting norms through learning mechanisms such as imitation may also appear to spread norms in agent societies.

### 2.2.2 Conventions vs. social norms vs. laws

It should be noted that researchers are divided on what the differences between a social norm and a convention are. Gibbs (1965, pg. 592) notes that *the terms “convention” and “custom” are frequently employed in the discussions of norms, but there does not appear to be any consensus in definitions of them beyond the point that they may not be sanctioned.*

We will assume that a convention is a common expectation amongst (most) others that an agent should adopt a particular action or behaviour (e.g. the convention in ancient Rome was to drive on the left). As conventions gain force, the violations of conventions may be sanctioned at which point a social norm comes into existence. For example, if driving on the right is sanctioned, the left-hand driving becomes a norm. A norm may become a law when it is imposed by an institution (e.g. laws that govern driving behaviour). Our view on the relationship between conventions, social norms and laws is given in Figure 2.4. Note that when conventions are established, they are not associated with sanctions (e.g. style of dress, dinner table etiquette). However, the conventions may become social norms once they are enforced by other agents due to the expectation of a particular behaviour. The enforcement happens at the peer-to-peer level (decentralized enforcement). When a norm has emerged in a society it may be institutionalized (i.e. it becomes a law (Finnemore and Sikkink, 1998)), which will then be formally enforced by a central authority (e.g. the government) that makes use of distributed legal entities such as a police department and the justice department.
2.2.3 Norm life-cycle

In the body of research literature on social norms, there is no unified view on how norms are created and spread in a society and various models of norms have been proposed (Elster, 1989b; Coleman, 1990; Opp, 2001; Horne, 2001; Finnemore and Sikkink, 1998; Bicchieri, 2006). According to Coleman (1990) “Norms are macro level constructs based on purposive actions at the micro level but coming into an existence through a micro-to-macro transition. Once in existence, they lead, under certain conditions, to actions of individuals (that is, sanctions or threat of sanctions) which affect the utilities and thus the actions of the individuals to whom the sanctions have been or might be applied”.

In the context of their study on international relations, Finnemore and Sikkink (1998) have proposed a three-stage model of the norm life-cycle: the norm emergence stage, the norm cascade stage and the norm internalization stage. The first stage is the norm emergence stage which is characterised by the persuasion of other agents to follow the norm by some norm entrepreneurs or norm innovators. Norm entrepreneurs are the innovators who think about new norms in a society (e.g. Henry Dunant the founder of Red Cross was the entrepreneur of the norm to treat wounded soldiers in a war as neutrals). Norm entrepreneurs attempt to convince a critical mass of norm leaders to embrace new norms. The motives of the entrepreneurs to come up with these norms include altruism and empathy. The second stage is the norm cascade stage characterised by the dynamics of imitation as the norm leaders attempt to socialise with other agents whom they might have influence over, so they might become followers. Followers may take up the norm because, following the norms might enhance their reputation and also their own esteem. They may also follow the norm because of the peer pressure from other followers. The third stage is the norm internalization stage where the norms are widely accepted by the agents in the society to the extent that they might be taken for granted (i.e. the norm following becomes an automatic task for the followers). An issue with the the internalized norms is that these norms can then become hard
to discern from some behavioural regularity as there is no discussion in the society about whether a norm should be followed.

Note that the researchers have broadly identified three phases of the norm life-cycle. In the context of describing the empirical works on norms we will revisit and extend the phases of the norm life-cycle model and discuss various mechanisms employed by the researchers in Section 3.1.

### 2.3 Normative multi-agent systems (NorMAS)

Research on norms in multi-agent systems is about two decades old (Shoham and Tennenholtz, 1992, 1995; Castelfranchi and Conte, 1995; Boman, 1999; Dignum, 1999; Conte et al., 1999). Norms have been of interest to multi-agent system (MAS) researchers as they help in maintaining social order (Conte and Dellarocas, 2001), facilitate cooperation (Axelrod, 1997) and coordination (Shoham and Tennenholtz, 1992; Walker and Wooldridge, 1995). Since norms enable smoother functioning of the societies by facilitating social order, MAS researchers have used this concept to build multi-agent systems. They have also investigated how norms may evolve in response to environmental changes.

#### 2.3.1 Definitions

The definition of normative multi-agent systems (NorMAS) as described by the researchers involved in the NorMAS 2007 workshop is as follows (Boella et al., 2008). “A normative multiagent system is a multiagent system organised by means of mechanisms to represent, communicate, distribute, detect, create, modify and enforce norms, and mechanisms to deliberate about norms and detect norm violation and fulfilment”.

In NorMAS, many research works treat norms as constraints (hard and soft) on actions that an agent can perform (Boella et al., 2009). Some researchers view norms as hard constraints where agents are not permitted to violate norms (Boman, 1997). A more common view is that norms are soft constraints where an agent has the ability to violate norms. One of the key attributes of an agent in a normative system is its “norm autonomy”.

According to Castelfranchi et al. (2000) a *norm autonomous* agent is
• able to know that a norm exists in the society and that it is not simply a diffuse habit, or a personal request, command or expectation of one or more agents

• able to adopt this norm impinging on its own decisions and behaviour

• able to deliberatively follow that norm

• able to deliberatively violate a norm in case of conflicts with other norms.

According to Verhagen (2000), norm autonomous agents are agents that choose which goals are legitimate to pursue, based on a given system of norms. The agent has the autonomy of generating its own goals and to choose which goals to pursue. Besides, the agent can also judge the legitimacy of its own and other agents’ goals. When a goal conflict arises, the agent may change its norm system, thereby changing priorities of goals, abandoning a goal, changing a goal, or generating another goal. Norm autonomous agents generate norms they can use to evaluate states of the world in terms of whether or not they could be legitimate interests.

These two definitions of norms are closely related. An autonomous agent first has to know about the existence of a norm and then adopt (or accept) the norm. It can then either follow or violate the norm based on its preferences. Verhagen’s definition includes the property that an agent can change its goals based on what the norm dictates. The definition also identifies the ability of the agents to generate norms based on observing the world states².

²A system under consideration can be in one of the possible states (i.e. world states) at any point of time. The state of the world at time $t$ is called as the world state. For example, the state of a simulated traffic system can be a) all agents driving on the left (state 1), b) all agent driving on the right (state 2) or c) no consensus in the driving direction (state 3). These world states are brought about by the actions of the individual agents. At time $t_1$, let us assume that 50% of the agents drive on the left and 50% of the agents drive on the right. In this case the state of the system (i.e. no consensus in the driving direction) can be inferred as non-normative because the driving actions have not converged to either right or left. However, at time $t_2$ the system may converge to driving on the left based on the recommendation of a leader agent. The norm of driving on the left can be inferred from the world state where every agent drives on the left.
2.3.2 Different views of norms in NorMAS

There are a number of different views of norms among normative multi-agent system researchers. Norms are treated as constraints on behaviour, goals to be achieved or as general obligations (Castelfranchi and Conte, 1995). Norms have also been treated as social laws, rights, contracts, protocols and commitments (López y López, 2003).

Some researchers have viewed norms as social laws (Shoham and Tennenholtz, 1995; Moses and Tennenholtz, 1995; Briggs and Cook, 1995). The two types of social laws that have been identified are the strict laws and flexible laws. Moses and Tennenholtz (1995) have argued that strict laws are norms. According to the strict law, an agent has a set of states out of which it is permitted to do certain actions in each state. It is assumed that the agents do not violate laws. This view severely restricts the autonomy of agents. In the flexible law theory proposed by Briggs and Cook (1995), the agents have a hierarchy of norms and the agents are given the flexibility of choosing the norm that best fits their current goal. They normally tend to obey the stricter norm that exists in the top of the hierarchy of norms. If they are unable to choose a norm from the top level of the norm-hierarchy (i.e. a stricter norm), they choose a relaxed norm lower in the hierarchy so that they can create a plan to achieve their goals.

Norms are also viewed as rights of agents. Some researchers (Norman et al., 1998; Alonso, 2002, 2004) have argued that norms give agents the right to perform an action without being punished or sanctioned. Norms as rights can be viewed as permissive norms (Alchourron, 1991). For example, only a leader agent L in a society S1 is allowed (or permitted) to send messages to agents A and B that are external to the system (say in society S2). The agent that is permitted to send a message (i.e. agent L) can choose to perform that action but is not required to do so (i.e. there is no expectation that agent L must send a message).

Norms have been used to specify contracts between agents (Dignum, 1999). Contracts describe obligations (what is expected of the agent) and authorizations (who can create obligations) between agents that are usually created explicitly and only hold for a limited time. These contracts are between an agent and a few other agents in the society. This is in contrast with conventions (or norms) which are between an agent and a society (i.e. all the agents in the society) and the private norms that are internal to an agent. The private norms can be
thought of as the personal values of the agent.

The normative behaviour of agents can be specified using protocols (Jennings, 1993). A protocol defines the set of rules an agent has to follow in a particular context. For example, a communication protocol defines the rules of interaction between agents. The protocol specifies actions that an agent is expected to perform or the messages it is expected to send to other agents. It also specifies the order in which the messages should be exchanged. Viewing norms as protocols has several weaknesses (Dignum, 1999). First, even though protocols allow agents to react in a predictable way for messages, if environments change, the agent cannot react to the changes if they are hard-wired not to violate norms. Second, an agent may be in a scenario in which it may have to violate a norm in order to satisfy a private goal. If norms have to be strictly followed (i.e. cannot be violated) as a result of following the protocol, the agent cannot satisfy its private goal. Dignum (1999) notes this inability to violate a norm raises the concern as to how autonomous the agents really are. Some researchers have addressed this issue by defining protocols for agents’ interactions and at the same time have allowed agents to violate the norms (e.g. based on the personality of the agents). These works also provide mechanisms to detect norm violations (Aldewereld et al., 2006b; Vázquez-Salceda et al., 2005).

Some researchers view that the existence of norms create commitments (Viganò et al., 2005; Jennings, 1993). For example, if norms exist in a society, an agent in that society is committed to abide by the norm. Commitments are viewed as pledges to undertake a specified course of action, while norms provide a means of monitoring commitments in changing circumstances (Jennings, 1993). Viganò et al. (2005) note that norms are not themselves commitments, but rules that manipulate commitments of the agents engaged in an interaction (i.e. norms create, modify and cancel commitments).

Some researchers have studied a general notion of social expectation (Cranefield and Winikoff, 2009). The term social expectation is an umbrella term that covers social expectations including expectations that arise from norms, promises and commitments. Since the focus of this thesis is on norms, we will focus on the expectations that arise because of norms.

Our view of norms in this thesis is that they are soft constraints on the behaviour of agents and the agents can autonomously decide whether to follow a norm.
2.3.3 Two branches of research in NorMAS

Researchers in normative multi-agent systems have been influenced from two different perspectives: philosophy of law (prescriptive approach) and conventionalistic approach (emergence approach) (Conte and Castelfranchi, 1999). Based on these two perspectives, research in normative multi-agent systems can be categorized into two branches. The first branch focuses on normative system architectures, norm representations, norm adherence and the associated punitive or incentive measures. The second branch is concerned with the emergence of norms.

Researchers from the philosophical background interested in the study of relationships between different mental attitudes of agents have formalized their theories, such as the Belief-Desire-Intention theory\(^3\). In the study of the prescriptive approach towards norms, researchers are interested in representing norms independent of the domain that is being studied. To that end researchers have used some form of logic, mainly deontic logic to represent norms. Deontic logic is the logic of prohibitions, obligations and permissions (von Wright, 1951). Prohibition norms apply to actions that an agent may perform or the undesired state of affairs that an action may bring about. For example an agent may be prohibited from littering in a park. Obligations are actions that an agent is expected to perform or a state it is expected to bring about. For example an agent may be obliged to tip a waiter in a restaurant. Permissions are used to indicate exceptions to a general rule or used in cases of uncertainty (Dignum, 1999). For example a student attending a lecture may be allowed to run out of a lecture theatre in the event of a fire. Deontic logic studies the relationship between these three concepts and also how violations and contrary-to-duty obligations are related (Boella et al., 2006). Norms can be represented as rules or conditions using deontic logic.

Even though this branch studies how norms are formalized and represented, it does not address the question of where the norm comes from (i.e. how a norm emerges in a society). Some researchers have proposed mechanisms by which norms can emerge in an agent society.

\(^3\)“Beliefs are statements of properties of the world the agent is situated in that can either be true or false; desires are states/situations that an agent prefers to bring about or actions that it wants to perform and intentions are those feasible actions, plans or desired situations that an agent has selected and committed to performing or achieving” (c.f. Dignum et al. (2002)).
(Verhagen, 2001; Sen and Airiau, 2007). Thus the second branch of research deals with the empirical approaches to norms. This branch of work differs from the first branch in terms of the different mechanisms explored by the researchers (e.g. leadership, reputation, machine learning, imitation) and the experiments that are conducted using these mechanisms. The emergence of norms is explored only by the research of this branch. We note that much of the work on norms in this branch does not make any distinction between conventions and norms (Shoham and Tennenholtz, 1992; Sen and Airiau, 2007; Villatoro and Sabater-Mir, 2009; Salazar et al., 2009) - both conventions and norms are included under the umbrella of norms. Most work on the emergence of norms (mainly conventions) are from a game-theory perspective (Axelrod, 1986; Shoham and Tennenholtz, 1992; Sen and Airiau, 2007).

Conte and Castelfranchi (1999) have worked on an integrated view of norms. Their work tries to bridge the gap between the prescriptive view of norms (first branch) and the emergence of conventions (second branch), using the cognitive abilities of an agent. They have proposed a logic-based framework to integrate these two perspectives. However, concrete implementations of this integrated approach are yet to be seen. The rest of this chapter describes the work that belongs to the first branch of research in NorMAS, while the next chapter describes research works on empirical works on norms.

### 2.3.4 Normative architectures

Several architectures have been proposed for the study of norms (López y López and Márquez, 2004; Boella and van der Torre, 2006). Researchers have created these different architectures to study and test their intuitions about norms (Neumann, 2010).

Boman (1999) views norms as rigid constraints of behaviour that cannot be violated. Norms are prescribed to the agents and these utilitarian agents use a variant of decision theory to identify the action that maximizes their expected pay-off (by minimizing the risks). Castelfranchi et al. (2000) are among the pioneers in the field to provide a framework for normative agents. They proposed an architecture for deliberative normative agents which had an explicit representation of norms. They argue that a cognitive agent needs an explicit representation to reason about norms unlike game theory based works where norms are not explicitly represented (see section 2.3.5 for details on how norms are represented). Accord-
According to Castelfranchi et al. (2000), deliberative normative agents are agents that have explicit knowledge about the enacted norms in a multi-agent environment and can make a choice on whether to obey the norms in specific cases. Upon identifying a norm, an agent should be able to deliberate how the norm will affect its desires and goals.

Boella and Lesmo (2001) have proposed an architecture for autonomous agents to decide whether to respect an obligation norm. These normative agents are modelled as a Belief-Desire-Intention (BDI) agent (Rao and Georgeff, 1995). The two types of agents in this architecture are the addressee agents who are the bearers of the norm (i.e. the subjects of the norm) and the defender agents who have the right to sanction the bearers if they violate the norm.

Broersen et al. (2001) have designed the BOID architecture which adds the obligation component to the BDI framework. The architecture describes how 16 types of conflicts between these four components can be resolved (e.g. conflict between beliefs and desires, conflicts between beliefs, desires and intentions). The authors have proposed several types of agents such as realistic, simple-minded, selfish and social agents. The conflicts between these four components of an agent can be resolved based on priority ordering of the preferences encoded in the type of the agent (e.g. in social agents, the obligations overrule desires). However, this architecture does not take into account the impact of sanctions on the part of the agent’s decision making.

Dignum et al. (2002) note that the “desire” component of the BDI framework has not been given proper consideration. Desires are often reduced to goals in other works. They note that desires and goals are important where other sources of motivations such as norms and obligations are present. They have developed an architecture for an agent called B-DOING which stands for beliefs, desires, obligations, intentions, norms and goals. The

---

4In the view of Dignum et al. (2002), the main difference between norms and obligations is that norms apply to a group or a society of agents while obligations are commitments of an agent to another agent in a society. They note that obligations arise because of an individual agent’s action (e.g. paying a seller agent after buying an item) and an agent’s interactions with different individuals may lead to distinct obligations. Thus the difference is the level at which the norm is modelled (i.e. between an agent and the society vs. between two or more agents in a society). We note that obligations that are expected of an agent in a society (e.g. an agent is obliged to tip in a society) are treated as norms (i.e. obligation norms) in many research works.
architecture is formalized using deontic logic.

Boella and van der Torre (2003) have proposed an architecture to detect norm violations based on a delegation model of social control. They have proposed a three level hierarchy for detecting norm violations. At the top level there exists a normative system which delegates the task of monitoring other agents to defender agents. The defender agents perform the task of monitoring whether norms are violated by the addressee (or bearer) agents.

Lopez et al. (2004) have designed an architecture for normative BDI agents. In this architecture the motivation of agents is introduced as a new mental attitude which is related to their autonomy (i.e. agents can choose goals based on their own preferences). This architecture is by far the most comprehensive in terms of considering different aspects of norms in the decision-making component of an agent. The architecture captures an agent’s normative goals, the addressees and beneficiaries of the norm, the context under which it holds, the exceptions to the norms (conditions under which it can be violated without punishments, i.e. “immunity” states), rewards and punishments. Besides motivations, three new mental attitudes are included: norm instances, intended norms and rejected norms. They have come up with several types (personalities) of normative agents.

Kollingbaum and Norman (2003a,b) have designed an architecture for an agent to resolve normative conflicts. For example an agent may be obliged and forbidden to perform an action at the same time, which leads to inconsistencies. Their approach can detect these inconsistencies. They have defined three consistency levels for obligations, namely strong consistency, weak consistency, and inconsistency (Kollingbaum et al., 2008). Conflicts are resolved by curtailing the scopes of norm influences. The scopes of norms are defined by the norm variables (Vasconcelos et al., 2009).

Boella and van der Torre (2006) have proposed a distributed architecture for normative agents which tries to bridge the gap between logic-based agent specification languages and programming languages used for implementation. In this architecture they have proposed components for counts-as conditionals, conditional obligations, and conditional permissions. They note that counts-as conditionals such as X counts as Y in context C can be used to link brute facts to institutional facts. For example, an agent not paying for an item it won in an auction (a brute fact) is considered as a violation from the institutional point of view (i.e. institutional fact). Brute facts arise from restrictive norms (e.g. an agent can or cannot
perform certain action), and the institutional facts arise from the regulatory norms (e.g. all agents in an institution ought to perform an action) (Searle, 1969).

Vázquez-Salceda et al. (2005) have extended the ISLANDER formalism (Esteva et al., 2002) that provides a formal framework for electronic institutions (e.g. an electronic auction house such as the Fishmarket (Noriega, 1997)) with normative reasoning capability. Their main interest stems from the fact that works on deontic logic have focused on providing a formal definition of norms including conditional and temporal aspects. However, they realize that it is important to provide not only a declarative meaning of norms but also an operational meaning (i.e. details required for implementing the norms in MAS). They have proposed a machine-readable format for expressing complex norms. The AMELI platform executes the ISLANDER specifications (Esteva et al., 2004). The extensions to the formalism include a more expressive mechanism to represent norms, and the distinct representations for abstract and concrete norms in an institutional setting. The language provides constructs to represent three deontic concepts, namely prohibitions, obligations, and permissions. Their main contribution is towards providing a framework for norm enforcement. They have used two mechanisms for the detection of violations. The institution can have a black list of actions to be checked, and an action alarm mechanism triggers an alarm when a given action attempts to start or is running or is done. Based on the alarms raised, sanctions are detected. They also propose mechanisms to represent norms with conditions and deadlines (norm activation and expiration). Recently the same set of researchers have demonstrated how concrete implementations of abstract norms specified by organizations can be provided (Aldewereld et al., 2010) by considering the context of the norm and the use of the counts-as relationship.

Sadri et al. (2006) describe a normative KGP agent (N-KGP) agent where KGP stands for three components of an agent namely knowledge, goals and plans. The architecture proposed for an agent uses abductive logic programming for the specification of normative concepts. An agent using the architecture can find when a norm holds and how to respond to a situation based on its preferences. The norms that apply to an agent are based on the roles it plays in a society.

For a detailed comparison of selected normative architectures, refer to Neumann’s article (Neumann, 2010). Neumann has categorized selected normative architectures based on a) the theoretical backgrounds of the architectures, b) the viewpoints of the architectures (single
agent vs. agent society), c) the reason for following a norm (deontic vs. consequentialistic view\(^5\)), d) the consideration of static vs. dynamic societies, and e) the ability to deal with conflicts.

### 2.3.5 Norm representation

Researchers have used both explicit and implicit data structures to represent norms. In the game theoretical work on norms, they are represented implicitly as simple data types. There is no explicit representation of a norm (i.e., an agent cannot communicate what the norm is). A reader of a paper has to infer what variables the norm is based on. For example in Axelrod's work (1986) on norms dealing with agents cheating in an exam situation, if all the agents abstain from cheating in an exam, it can be inferred that each agent has the norm against cheating. The variable to cooperate (i.e., abstaining from cheating) is not explicitly represented in such an agent, but it can be inferred by the reader.

Several researchers have used deontic logic to define and explicitly represent norms (Wieringa and Meyer, 1994; Jones and Sergot, 1993; García-Camino et al., 2006b; Boella and van der Torre, 2006). In deontic logic, norms of permissions, prohibitions and obligations are studied. For example, \( P_A(litter|home) \) implies that agent A is permitted to litter given that she is at her home. \( F_A(litter|park) \) implies that agent A is prohibited from littering a park. \( O_A(pay|win) \) implies that agent A is obliged to pay after winning an item in an auction.

Some researchers have used decision trees (Boman, 1999; Verhagen, 2001; Lotzmann and Möhring, 2009) to represent norms. Norms represented as protocols are specified using some form of temporal logic (Aldewereld et al., 2006b; Endriss, 2005; Cranefield, 2007). UML diagrams (Pitt et al., 2001; da Silva and Braga, 2009) and Petrinets (Fix et al., 2006) have also been used to represent normative protocols. Norms are implemented as rules in rule-based systems (García-Camino et al., 2005; Cranefield, 2007). Silva (2008) has proposed a mechanism for transforming a norm specification into implementation (a set of rules) which are used to govern the behavior of the agents. Artikis (2003) has used the C+ language

\(^5\)Deontic view of norms advocates that norms are in itself a reason for action. On the other hand in the consequentialist view, actions are judged by their consequences (e.g. based on the utility of the actions).
and the event calculus for creating executable specifications of open norm governed systems. We will revisit the topic of norm representation in the context of discussing empirical works on norms in Section 3.3.1.5.

### 2.3.6 Norm enforcement

Apart from the architectures that have in-built mechanisms for checking norm violations and enforcing norms, other works have proposed mechanisms for norm enforcement (de Pinninck et al., 2008; Fix et al., 2006). A recent development is the research on emotion-based mechanisms for norm enforcement (Staller and Petta, 2001; Scheve et al., 2006; Fix et al., 2006). Upon internalization, emotions generated by agents when they violate a norm serves the purpose of enforcing the norm in an agent.

When agents violate a norm, other agents may abstain from interacting with this agent and so it becomes an outcast. This ostracism-based mechanism has been used to dissuade norm violators in an agent society (de Pinninck et al., 2008). Some researchers have employed governor agents to monitor norm compliance (García-Camino et al., 2006a).

Whereas most works consider sanctions for norm enforcement, López y López et al. (2002) have also considered a reward-based mechanism for norm enforcement. Criado et al. (2010) have extended the work of López y López et al. (2002) and have considered the role of rewards in enforcing a norm.

### 2.3.7 Discussion

Note that only a few of these architectures have been implemented as a normative system, even at the prototype level (Broersen et al., 2001; García-Camino et al., 2006b; Sadri et al., 2006). One of the key components of a normative agent missing in the works described above is the ability for an agent to recognize the norm of a society. Most architectures assume that agents somehow know what the norms of the society are. For example the norms are given to agents by the designers in the form of protocols (Aldewereld et al., 2006b; Viganò et al., 2005). In some cases (Sadri et al., 2006), researchers note that the norms are dictated by roles that the agents play. However, none of the architectures described so far propose how an agent can identify norms in an open multi-agent society.
In an open agent society an agent may join and leave different societies, and the norms of these society may change over time. Additionally, the agent may play different roles in different societies. So it may not be desirable to assign norms to agents that operate in open agent societies \textit{a priori} (i.e. during design time). Instead, agents should be endowed with mechanisms for identifying or recognizing the norms in the society. In essence, there is a need for an architecture for an agent to identify norms in an agent society. This is a focus of this thesis and is addressed in chapters 6 to 9.

2.4 Summary

This chapter provided an overview of software agents and multi-agent systems. It also provided an introduction to norms and described how the concept of norms is being studied by the researchers in the normative multi-agent system community. In particular, it provided an overview of the research works on normative architectures, norm representation and norm enforcement. The next chapter provides an overview of the empirical works on norms in NorMAS.
Chapter 3

Empirical approaches to the study of norms

Several researchers in multi-agent systems have adopted empirical approaches to the study of norms using multi-agent based simulations. Multi-agent based social simulation is a branch of multi-agent systems where simulations of agents endowed with social concepts such as norms, cooperation, and competition are studied (Davidsson, 2002). These studies are important, as they enable researchers to gain deeper understanding of the phenomena that is being studied (e.g. crowd behaviour in the event of fire in an auditorium) by allowing them to experiment with different types of scenarios. This can help predict the behaviour of the system and also identify parameters of the system that result in “interesting” behaviours. Researchers studying norms have worked on simulations to study human behaviour (Axelrod, 1986, 1997). They have also studied how norms can facilitate social control in artificial agent societies (López y López, 2003) and also how they may emerge (Sen and Airiau, 2007; Villatoro et al., 2009).

The objectives of this chapter are four-fold. First, based on the empirical approach to the study of norms, we identify and discuss a norm life-cycle model consisting of five phases. Second, different mechanisms employed by the research works on norms in each of the five phases are discussed. Third, based on the discussion of different research works, we identify important characteristics of an individual agent and a society. We also compare research works based on these characteristics. Fourth, we discuss the contributions of this thesis in...
the context of the categorization described in this chapter.

3.1 Developmental phases of norms based on empirical studies

Broadly, from the view point of the society, the three important stages of norms are the formation stage, propagation stage and the emergence stage. Researchers employing empirical approaches to norms have investigated various mechanisms associated with norms with each of these stages. Mechanisms employed in the norm formation stage aim to address how agents can create norms in a society and how individual agents can identify the norms that have been created. Mechanisms used in the norm propagation stage aim to explain how norms might be spread and enforced in the society. The emergence stage is characterized by determining the extent of the spread of a norm in the society.

Figure 3.1 shows an overview of these three stages from the view point of a society (or a bird’s eye view). The larger green circles represent agent societies. The smaller blue circles represent agents and the red squares inside the blue circles represent norms. The bi-directional arrows represent interactions between agents. The first green circle depicts an agent society in the norm formation stage. This society has nine agents. An agent in this society has created a norm (the blue circle that has a solid red square inside). The second green circle shows the agent society in its norm propagation stage. The norm from the agent in the middle of the society has propagated to other agents it is connected to (i.e. the agents it interacts with). The last green circle shows the agent society in the norm emergence stage assuming that the threshold for norm emergence from the viewpoint of the external observer of the society is 75%.

Based on these three important stages of norms, we identify five phases (i.e. expanded stages) of the norm life-cycle which are norm creation, identification, spreading, enforcement and emergence as shown in Figure 3.2. We use the term norm life-cycle to capture the important aspects of a norm from its creation to its establishment and de-establishment in the society. One of the limitations of the well-known life-cycle model by Finnemore and Sikkink (1998) is that it does not explicitly identify the operational phases within a given
Figure 3.1: Three main stages of the norm life-cycle

stage of a norm\textsuperscript{1}.

Figure 3.2 shows the five phases of the norm life-cycle on the left and the mechanisms investigated by researchers for each of the phases on the right. We have categorized the mechanisms used in the empirical works on norms into nine main categories (marked with a * in Figure 3.2).

This section provides an overview of the five developmental phases of norms, and the next section provides a detailed discussion of the mechanisms studied by researchers in each of these phases.

3.1.1 Norm creation

The first phase of the life-cycle model is that of norm creation. The norms in multi-agent systems are created by one of the three approaches. The three approaches are a) a designer specifies norms (off-line design) (Conte and Castelfranchi, 1995), b) a norm-leader specifies norms (Boman, 1999; Verhagen, 2001), c) a norm-entrepreneur considers that a norm is good for the society (Hoffmann, 2003). In the off-line design approach, norms are designed off-line, and hard-wired into agents. This approach has been used by researchers to investigate norms that might be beneficial to the society as a whole using social simulations. In leadership approach, some powerful agents in the society (the norm-leaders) create a norm.

\textsuperscript{1}For example, the norm formation stage may consist of two phases, the norm creation phase and the norm identification phase. A leader agent may propose a new norm (i.e. the norm creation phase). A new agent joining the society may identify the norm by observing the norm being followed in the society (i.e. the norm identification phase).
The leadership approach can be based on authoritarian or democratic leadership. The leader can provide these norms to the follower agents (Boman, 1999; Verhagen, 2000). In the entrepreneurship approach to the creation of norms, there might be some norm entrepreneurs who are not necessarily the norm leaders but create a proposed norm\(^2\). When an agent creates a new norm it can influence other agents to adopt the norm (Finnemore and Sikkink, 1998; Hoffmann, 2003).

### 3.1.2 Norm identification

If a norm has been created in the society using one of the explicit norm creation approaches discussed in Section 3.1.1, then the norm may spread in the society (e.g. through communication). However, if the norms have not been explicitly created (i.e. norms cannot be explicitly stated because they have not been created), then an agent will need a mechanism to identify norms from its environment based on the interactions with other agents. In game-theory based empirical works (Shoham and Tennenholtz, 1992; Sen and Airiau, 2007), agents have a limited number of actions that are available, and they choose the action that maximizes their utility based on some learning mechanism such as imitation, machine

---

\(^2\)At this stage, the norm exists only in the mind of one agent (i.e. it is a personal norm). It hasn’t become a social norm that is accepted by other agents.
learning or data-mining. These works do not consider the cognitive abilities of agents in identifying norms.

The second approach to norm identification considers the cognitive capabilities of an agent to infer what the norms of the society are (Andrighetto et al., 2007, 2010a). In the cognitive approach, one or more cognitive agents in a society may come up with norms based on the deliberative processes that they employ (Andrighetto et al., 2007, 2010a). In this approach the other agents have the cognitive ability to recognize what the norms of a society are based on the observations of interactions. Agents have normative expectations, beliefs and goals. It should be noted that the norms inferred by each agent might be different (as they are based on the observations that the agent has made). Thus, an agent in this model creates its own notion of what the norms are based on inference.

3.1.3 Norm spreading

Norm spreading relates to the distribution of a norm among a group. Once an agent knows what the norm in the society is (i.e. either based on norm creation or identification), several mechanisms help in spreading the norms such as leadership, entrepreneurship, cultural, and evolutionary mechanisms (explained in detail in Section 3.2). It should be noted that for an external observer, agents identifying norms through learning mechanisms such as imitation⁴ appear to spread norms in agent societies.

3.1.4 Norm enforcement

Norm enforcement refers to the process by which norm violators are discouraged through some form of sanctioning and norm followers are encouraged through rewards. A widely used sanctioning mechanism is the punishment of a norm violator (e.g. monetary punishment which reduces the agent’s fitness or a punishment that invokes emotions such as guilt and embarrassment). Reputation mechanisms have also been used as sanctions, such as where an agent is black-listed for not following a norm. The process of enforcement helps to sustain norms in a society.

⁴Societies that employ imitation mechanisms can have co-existing norms (Epstein, 2001) for the same context since agents can imitate different actions of agents.
Note that enforcement of norms can influence norm spreading. For example, when a powerful leader punishes an agent, others observing this may identify the norm. Hence, the norm can be spread. Norms can also be spread through positive reinforcements such as rewards. Some researchers have considered enforcement as a part of the spreading mechanism (Axelrod, 1986) (see Section 3.2.6 for a detailed discussion).

3.1.5 Norm emergence

The fifth phase is the norm emergence phase. We define norm emergence to be reaching some significant threshold in the extent of the spread of a norm; that is a norm is followed by a considerable proportion of an agent society and this fact is recognised by most agents. For example, a society could be said to have a norm of gift exchange at Christmas if more than \( x\% \) of the population follows such a practice. The value of \( x \) varies from society to society and from one kind of norm to another. The value of \( x \) has varied from 35 to 100 across different empirical studies of norms (see Table 3.3).

Emergence can be detected either from a global view of the system or through a local view\(^4\) of an agent (e.g. an agent might only see agents that are one block away on all directions in a grid environment\(^5\)). Spreading of norms with or without enforcement can lead to emergence. Once a norm has emerged, the process can continue when an entrepreneur or a leader comes up with a new norm that replaces the old one. This is indicated by a dotted arrow in Figure 3.2.

The adoption of a norm may decrease in a society due to several reasons. A norm that has emerged may lose its appeal when the purpose it serves does not hold or when there are not enough sanctions or rewards to sustain the norm or when other alternate effective norms emerge. Note that the model presented here is from a bird’s-eye view (i.e. an external agent that observes the society). An external agent will able able to observe the norm establishment and de-establishment in the society based on the emergence criterion (i.e. the extent of spread of the norm).

\(^4\)Note that the norms observed in the local view could be different from the norms that can be observed from a global view.

\(^5\)Agents in a particular environment can be connected to one another using one of many topologies such as regular, small-world, scale-free and fully-connected topologies (Mitchell, 2006).
3.1.6 Consideration of network topologies

An important attribute of the research works on norms is the consideration of network topology. The underlying interaction topology of agents has an impact on all phases of norm development. For example the interactions between a leader and his followers have an implicit network topology (i.e. fully-connected network) which governs how norms created by the leader may spread and may lead to norm emergence in an agent society. Hence the consideration of network topology is included as one of the nine main categories. The network structure of the society can either be static or dynamic (i.e. can evolve due to agents joining and leaving).

3.1.7 Discussion

The life-cycle that we have presented is similar to Finnemore and Sikkink’s model (1998) described in Section 2.2.3. As the reader may observe, their model is a subset of the life-cycle model that we have proposed. Finnemore and Sikkink’s model caters only for the entrepreneurial approach for norm creation and the imitation approach for norm spreading. However, in our life-cycle model more mechanisms are brought under each of the phases. For example mechanisms based on emotions, culture and evolution are included in our model.

Another distinction between the models is the viewpoint of the life-cycle. Finnemore and Sikkink’s model includes the social mechanisms employed by human agents (e.g. entrepreneurship, imitation, reputation). Our model is based on a socio-computational viewpoint which includes modeling social mechanisms from a computational viewpoint and also studying pure computational techniques in the empirical study of norms which are enacted by software agents. For example offline design approaches and machine learning mechanisms are only applicable to our model. These mechanisms can be used to study phenomena which may otherwise be difficult without the help of computational modeling and abstractions.

Unlike other categories such as off-line design and learning, the consideration of network topology is not strictly a mechanism that is relevant to one or two phases of norm development but a consideration that may have an impact on any phase of norm development (i.e. network topology is an orthogonal consideration). However, as we believe network topology is an important aspect in the study of norms we have included it as one of the categories in our categorization (see Figure 3.2).
While addressing how norms are created and spread, the proposed life-cycle model can also accommodate the process of norm change. A norm entrepreneur can come up with a modified norm which can be spread by one of the spreading mechanisms, which may lead to the replacement of the older norm with a new one, or cognitive agents might notice a change in norms due to a change in the society’s membership.

Table 3.1 provides an overview of the contributions of selected empirical research works to these different phases of the norm life-cycle. It should be noted that not all phases of the norm life-cycle have been taken into account by most works, and some works make use of more than one mechanism in a single phase. We have also shown the mechanisms employed by this thesis in the last two rows of Table 3.1.

Table 3.1: Mechanisms used in different phases of the norm life-cycle

<table>
<thead>
<tr>
<th>Empirical works</th>
<th>Creation</th>
<th>Identification</th>
<th>Spreading</th>
<th>Enforcement</th>
<th>Emergence</th>
</tr>
</thead>
<tbody>
<tr>
<td>Axelrod (1986)</td>
<td>-</td>
<td>-</td>
<td>Evolution</td>
<td>Sanction</td>
<td>Yes</td>
</tr>
<tr>
<td>Shoham and Tennenholtz (1992)</td>
<td>-</td>
<td>Machine learning</td>
<td>-</td>
<td>-</td>
<td>Yes</td>
</tr>
<tr>
<td>Conte and Castelfranchi (1995)</td>
<td>Off-line</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>Walker and Wooldridge (1995)</td>
<td>-</td>
<td>Machine learning</td>
<td>-</td>
<td>-</td>
<td>Yes</td>
</tr>
<tr>
<td>Shoham and Tennenholtz (1995)</td>
<td>Off-line</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
</tr>
</tbody>
</table>

*Yes : considered in the model, - : not considered/specified*

Continued on next page

42
<table>
<thead>
<tr>
<th>Empirical works</th>
<th>Creation</th>
<th>Identification</th>
<th>Spreading</th>
<th>Enforcement</th>
<th>Emergence</th>
</tr>
</thead>
<tbody>
<tr>
<td>Castelfranchi et al. (1998)</td>
<td>Off-line</td>
<td>-</td>
<td>-</td>
<td>Reputation</td>
<td>-</td>
</tr>
<tr>
<td>Verhagen (2001)</td>
<td>Leadership</td>
<td>-</td>
<td>Leadership</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>Epstein (2001)</td>
<td>-</td>
<td>Imitation</td>
<td>Cultural transmission</td>
<td>Sanction</td>
<td>Yes</td>
</tr>
<tr>
<td>Flentge et al. (2001)</td>
<td>-</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Hales (2002)</td>
<td>Off-line</td>
<td>-</td>
<td>-</td>
<td>Reputation</td>
<td>-</td>
</tr>
<tr>
<td>López y López et al. (2002); López y López (2003)</td>
<td>Off-line</td>
<td>-</td>
<td>-</td>
<td>Sanction and reward</td>
<td>-</td>
</tr>
<tr>
<td>Chalub et al. (2006)</td>
<td>-</td>
<td>Machine learning</td>
<td>Evolution, network topology</td>
<td>-</td>
<td>Yes</td>
</tr>
<tr>
<td>Fix et al. (2006)</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>Emotion</td>
<td>-</td>
</tr>
<tr>
<td>Pujol (2006)</td>
<td>-</td>
<td>Machine learning</td>
<td>network topology</td>
<td></td>
<td>Yes</td>
</tr>
<tr>
<td>Sen and Airiau (2007)</td>
<td>-</td>
<td>Machine learning</td>
<td>-</td>
<td>-</td>
<td>Yes</td>
</tr>
</tbody>
</table>

Yes: considered in the model, -: not considered/specified
Continued on next page
3.2 Categorization of empirical works on norms

In this section we categorize empirical works on norms into nine main categories as shown in Figure 3.3 (marked with a *)\(^7\). For each of these categories, we provide a brief description and discuss a few key papers. It should be noted that some papers have made use of mechanisms that fall under more than one category (e.g. Axelrod’s work (1986)), and also a category may contribute to different phases of the norm life-cycle (i.e. leadership and entrepreneurship mechanisms can be used to facilitate norm creation and spreading).

3.2.1 Off-line design approaches

Off-line design models are characterised by the agents of the society possessing explicit knowledge of the norms. The intention of this approach is to seed agents with norms and

---

<table>
<thead>
<tr>
<th>Empirical works</th>
<th>Creation</th>
<th>Identification</th>
<th>Spreading</th>
<th>Enforcement</th>
<th>Emergence</th>
</tr>
</thead>
<tbody>
<tr>
<td>Andrighetto et al. (2010a); Campenní et al. (2009)</td>
<td>-</td>
<td>Cognition, imitation</td>
<td>-</td>
<td>-</td>
<td>Yes</td>
</tr>
<tr>
<td>Chapters 4 &amp; 5</td>
<td>-</td>
<td>Machine learning</td>
<td>Leadership, network topology</td>
<td>-</td>
<td>Yes</td>
</tr>
<tr>
<td>Chapters 6-9</td>
<td>Off-line</td>
<td>Cognition, Data mining</td>
<td>-</td>
<td>Sanction</td>
<td>No</td>
</tr>
</tbody>
</table>

*Yes : considered in the model, - : not considered/specified

\(^7\)Even though we distinguish norms from conventions (see Section 2.2.2), for the purpose of categorization, we have incorporated both conventions and norms under the umbrella of norms.

\(^8\)These are the same nine categories shown in Figure 3.2.
compare how the society performs when the whole society possesses certain norms as opposed to agents behaving strategically (without possessing the notion of norms). One of the well-known works on norms specified by the designer is by Shoham and Tenenbholz (1995), who experimented with norms associated with traffic. Several other researchers (Walker and Wooldridge, 1995; Conte and Castelfranchi, 1995; Hales, 2002) have experimented with an off-line design approach borrowing the basic experimental set-up proposed by Conte and Castelfranchi (1995). Conte and Castelfranchi (1995) have shown using their simulation experiments what the function of a norm is in the context of agents finding food in a grid environment characterised by simple rules for movement and food collection. An agent’s strength increases when it consumes food and its strength decreases when it moves from one cell to another. This work compared the utilitarian strategy with the normative strategy and
showed that norms reduce the aggression level of the agent (when a finders-keepers norm is followed) and also increase the average strength of an agent.

The work of Conte and Castelfranchi (1995) assumes that an agent society is either made up of strategic agents or normative agents. López y López et al. (2002) have extended the notion of off-line design by experimenting with an agent society by varying the types of agent personalities.

Discussion - Off-line design models are best suited for studying and comparing different normative schemes in a closed agent society. However, agents inhabiting open and distributed environments may not have the privilege of consulting a designer. Walker and Wooldridge (1995) note the following about the off-line design of norms. “[This] approach will often be simpler to implement and might present the designer with a greater degree of control over system functionality. However, there are a number of disadvantages with this approach. First, it is not always the case that all the characteristics of a system are known at design time; this is most obviously true of open systems. . . . . Secondly, in complex systems the goals of agents . . . might be constantly changing. To keep reprogramming agents in such circumstances would be costly and inefficient. Finally, the more complex a system becomes, the less likely it is that system designers will be able to design effective social laws”. Some researchers have used this approach to compare the performance of a normative system with a non-normative one (Conte and Castelfranchi, 1995).

Another limitation of the off-line design mechanism is that it assumes all the agents in the society will follow a norm (e.g. the finders-keepers norm) which might not be realistic. Although this is possible for a well entrenched norm in a society, open societies might have different competing norms that are present at a given point of time.

3.2.2 Leadership and entrepreneurship mechanisms

Social power plays an important role in societies in establishing order and enabling smoother functioning. Several researchers of normative multi-agent systems have focused on the notion of power (Castlefranchi et al., 1992; Jones and Sergot, 1996; Castelfranchi, 2002) such as institutional power. López in her thesis on social power and norms notes that the social
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 (López y López, 2003). Sources of power could motivate, encourage, or persuade their followers to take up a particular norm (the leadership approach) or coerce them to adopt a particular norm based on sanctions (the punishment approach). Researchers have used leadership approaches for norm creation and spreading and have also experimented with sanction approach for norm enforcement (see Section 3.2.6).

Leadership mechanisms are based on the notion that there are certain leaders in the society, who provide advice to the agents in the society. The follower agents seek the leaders’ advice about the norm of the society.

Boman (1999) has used a centralised approach, where agents consult with a normative advisor before they make a choice on actions to perform. Verhagen (2001) has extended this notion of normative advice to obtaining normative comments from a normative advisor (e.g. the leader of the society) on an agent’s previous choices. The choice of whether to follow a norm and the impact of the normative comment on an agent are determined by the autonomy of the agent. Once an agent decides to carry out a particular action, it announces this decision to all the agents in the society, including the leader of the society, and then carries out that action. The agents in the society can choose to send their feedback to this agent. When considering the received feedback, the agent can choose to give a higher weight to the feedback it received from the leader agent. Verhagen has experimented with the internalization of norms in an agent society. Internalization refers to the extent to which an agent’s personal model of a norm matches the group model of a norm.

Hoffmann (2003) has experimented with the notion of norm entrepreneurs who think of a norm that might be beneficial to the society. An entrepreneur can recommend a norm to a certain percentage of the population (e.g. 50%) which leads to varying degrees of establishment of a norm. This model assumes that the agents in the society are willing to converge towards a norm. If their current norm deviates from the group norm (which is published by a centralised mechanism), an agent decrements the usefulness of its norm and may even choose another norm from the list of norms (or rules) available from a pool. Hoffmann’s experiments explore the entrepreneurial norm dynamics and provide some initial evidence for Finnemore and Sikkink’s norm life-cycle model (1998). Some shortcomings of this model
as acknowledged by the authors include the assumption of a single norm entrepreneur, the lack of communication between agents about norms, and the use of a centralised monitor to compute consensus.

**Discussion** - The leadership models assume that a powerful authority is present in the society and all agents in the society acknowledge the power of such agents. Both centralised and distributed notions of norm spreading using *power* have been employed. The centralised approach is suitable for closed societies. However, this might not work well for open, flexible and dynamic societies. Distributed approaches for norm spreading and emergence are promising because the computational costs required to spread, monitor and control a norm are distributed across the individual agents.

Another criticism of the centralised leadership mechanism in empirical models is that it assumes that an *all knowledgeable* authority is present in the society. Though this might be how some aspects of human societies are modelled (e.g. an institution), it would be challenging to model an agent (human or artificial) that might think of possible norms and recommend the one that might be the best so that others could use them.

### 3.2.3 Learning mechanisms

Three types of learning mechanisms have been employed by researchers: imitation, machine learning and data mining.

#### 3.2.3.1 Imitation mechanisms

The philosophy behind an imitation mechanism is *When in Rome, do as the Romans do* (Epstein, 2001). These models are characterised by agents mimicking the behaviour of what the majority of the agents do in a given agent society (following the crowd). Epstein’s main argument (2001) for an imitation mechanism is that individual thought (i.e. the amount of computing needed by an agent to infer what the norm is) is inversely related to the strength of a social norm. This implies that when a norm becomes entrenched the agent can follow it without much thought. Epstein has demonstrated this in the context of a driving scenario in which agents can observe each other’s driving preference (left or right) based on a certain
observation radius $r$. If the agent sees more agents driving on the right within the observation radius, it changes to the right. When a norm is established, the observation radius becomes one (i.e. the agent looks at one agent on its right and left to update its view about the norm). Other researchers have also experimented with imitation models (Pujol, 2006; López y López, 2003; Andrighetto et al., 2010a).

**Discussion** - Imitation might be a good mechanism when agents want to avoid the cost of reasoning about what the norm of the society is. An agent using the imitation model is not involved in the creation of the norm. It is just a part of the norm spreading effort based on copying others without much thought. Other researchers have noted that an imitation approach cannot bring about the co-existence of multiple norms in a society (Campenní et al., 2009; Nakamaru and Levin, 2004). This issue has to be scrutinised further because Epstein has shown that imitation can result in the creation of certain pockets of local consensus on norms even though a global consensus has not been arrived at. Another issue for debate is whether imitation-based behaviour (solely) really leads to norms as there is no notion of generalized expectation.

### 3.2.3.2 Works based on machine learning

Several researchers have experimented with agents finding a norm based on learning on the part of an agent (Shoham and Tennenholtz, 1992; Walker and Wooldridge, 1995; Sen and Airiau, 2007).

Shoham and Tennenholtz (1992) were the first in multi-agent systems research to experiment with norm emergence. They viewed a norm as a social law which constrains actions or behaviours of the agents in the system. They used a mechanism called co-learning which is a simple reinforcement learning mechanism based on a Highest Cumulative Reward (HCR) rule for updating an agent’s strategy when playing a simple coordination game and a cooperation game (prisoner’s dilemma). According to this rule, an agent chooses the strategy\(^9\) that has yielded the highest reward in the past \(m\) iterations. The history of the strategies chosen and the rewards for each strategy is stored in a memory of a certain size (which can be var-

---

\(^9\)The strategies are actions an agent chooses to perform such as cooperate and defect.
ied). They experimented with the rate at which the strategy is updated (after one iteration, two iterations, three iterations, etc.). When the frequency of update decreases, convention emergence decreases. They experimented with flushing the memory of the agent after a certain number of iterations and retaining only the strategy from the latest iteration. They found that when the interval between memory flushes decreases, the efficiency of the convention emergence decreases. One limitation of this model is that the agents do not have knowledge of the function of the norm (they are merely following an algorithm). Also, no notion of sanction or reward for violating or following the norms is included.

The experimental model of Walker and Wooldridge (1995) is based on the work done by Conte et al. (1995) where agents move about a grid in search of food. They experimented with 16 mechanisms for norm emergence. Their model used two parameters: the majority size and the strategic update function. Each of these parameters can be varied across four values. 16 experiments were based on the size of the majority (simple, double, quadruple, dynamic) and the nature of the update function (using majority rule, memory restart, communication type and communication on success). The model’s novelty is that it considered the role of memory and communication mechanisms. Also, the agents learn from their own interaction experience with other agents. Visibility of an agent is restricted to a particular region (or neighbourhood) which is governed by the “extroversion radius” (similar to the one suggested by Shoham and Tennenholtz (1997)). They concluded that further experiments should be done to enhance the understanding of this complex topic. They noted that the role played by social structure (i.e. the network topology) and communication should be considered in future work.

Sen and Airiau (2007) proposed a mechanism for the emergence of norms through social learning. They experimented with three reinforcement learning algorithms and the agents learned norms based on private local interactions. They observed that when the population size is larger, the norm convergence is slower, and when the set of possible action states is larger, the convergence is slower. They also studied the influence of the dynamic addition of agents with a particular action state to a pool of existing agents, as well as norm emergence in isolated sub-populations.
3.2.3.3 Data mining mechanism

Agents can also use a data mining approach to identify norms in agent societies. Agents in open agent societies can identify norms based on what they infer based on their observations of the society. The repository of an agent’s observations can then be mined for patterns of behaviour. There has been a proposal of an agent architecture for normative systems to employ data mining for citizens of a country to find information and norms from official documents (Stane and Zytniewsk, 2009). However, the work does not describe what types of norms are discovered and also the mechanisms used in the identification of norms.

This thesis presents an architecture for norm identification which employs association rule mining, a data mining approach. Mechanisms for identifying two types of norms, prohibition norms and obligations norms are studied. Our investigations on norm identification are discussed in Chapters 6 to 9 of this thesis.

3.2.3.4 Discussion

Learning mechanisms employ a particular algorithm to identify a strategy that maximizes an agent’s utility and the chosen strategy is declared as the norm. Since all agents in the society make use of the same algorithm, the society stabilises to a uniform norm. Agents using this approach cannot distinguish between a strategy and a norm. These agents accept the strategy that maximizes its utility as its norm. However, the agents do not have a notion of normative expectation associated with a norm (i.e. when agents expect certain behaviour on the part of other agents). This issue can be addressed by making an agent deliberate about a learnt norm (i.e. how the newly learnt norm might affect its desires, goals and plans).

Another weakness of these works is that agents lack an explicit representation of norms. Hence, they are not able to communicate norms with others. This weakness has been noted by several researchers (Castelfranchi et al., 2000; Sartor, 2001). Although the issue of norm communication between agents has been considered by an early work using an appropriate communication protocol (Walker and Wooldridge, 1995), explicit norm representation has not been considered by most empirical works on norms.
3.2.4 Cognitive approach

Researchers involved in the EMIL project (Andrighetto et al., 2007) have developed a cognitive architecture for norm emergence to explore how the mental capacities\(^{10}\) of agents play a role in the emergence of norms. We note that this is the only architecture that has been developed to study the emergence of norms.

The EMIL project aimed to deliver a simulation-based theory of norm innovation, where norm innovation is defined as the two-way dynamics of an inter-agent process and an intra-agent process. The inter-agent process results in the emergence of norms where the micro interactions produce macro behaviour (norms). The intra-agent process refers to what goes on inside an agent’s mind so that they can recognise what the norms of the society are. This approach is different from learning models, as the agents in the cognitive approach are autonomous and have the capability to examine interactions between agents and are able to recognise what the norms could be. The agents in this model need not necessarily be utility maximizing like the ones in learning models. The agents in this model will have the ability to filter external requests that affect normative decisions and will also be able to communicate norms to other agents. Agents just employing learning algorithms lack these capabilities.

Andrighetto et al. (2010a) have demonstrated how the norm-recognition module of the EMIL-A platform answers the question “how does an agent come to know of what a norm is?”. In particular they have experimented with an imitation approach versus the norm-recognition approach that they have developed. The norm recognition module consists of two constructs: the normative board and a module for storing different types of messages (which the authors call “modals”) that can be used to infer norms. The messages that are exchanged between

\(^{10}\) The mental capacity of an agent describes its ability to store internal representations of the external material world (e.g. beliefs about the world), create its own goals and plans and its ability to perform reasoning (also called as the agent’s mind) (Jonker and Treur, 1997). Let us assume that two teams of robotic agents are playing a RoboCup soccer match. Consider a scenario where robot R1 playing for team R observes robot R2 passing the ball to another robot R3. R1 creates a symbolic representation of what it observed as \textit{passBall (R2,R3)} and adds it to its belief set. In this case R1 has created a belief (an internal representation) of an interaction that happened in the external world. Robot R1 will then reason about which action to choose based on its beliefs, goals and plans (e.g. run towards R3 to shield it from opponents, move forward in anticipation that R3 will kick the ball towards it).
agents can be of five different types (e.g. the deontics modal refers to partitioning situations as either acceptable or unacceptable). The normative board consists of normative beliefs and normative goals, which are modified based on the messages received. They have shown that norm recognisers perform better than social conformers (imitating agents) because the recognisers were able to identify a pool of potential norms while the imitators generated only one type of norm.

The limitation of this approach is that agents just observe actions performed by other agents. In practice they should be able to learn from their own experience as well. Perhaps their own experience can be given a higher weight. At present, agents in this model do not have the capability of violating norms and hence there are no costs associated with sanctions. The authors note this as a potential extension.

In the work done by Campenní et al. (2009), the experimental set-up used by Andrighetto et al. (2010a) was extended\textsuperscript{11}. In the original set-up, agents moved across four different contexts. In each context an agent chose one of the three available actions. In each context, two of the three actions were unique to that context, while one action was common to all four contexts. So, there were nine actions in total. In an extension of this work (Campenní et al., 2009), barriers were used to prevent agents moving from one context to another. The agents were restricted to one of the contexts. The authors then compared how norms emerged with and without the barriers. It was noted that without the barriers, only one type of norm emerged (all the agents converged to the action that was common to all contexts), but when the barrier was used, three possible normative beliefs were generated (one for the common action, and two for two other actions). This result is interesting because, one would think, five different types of normative beliefs would be generated (one for the common context and one for each of the four contexts). However, it is not clear from the paper why only a total of three normative beliefs were created.

\textbf{Discussion} - Cognitive mechanisms are promising because agents with this type of mechanism have the notion of normative expectation. This mechanism focuses on what goes on inside the mind of an agent to infer norms (i.e. the computational machinery used by the

\textsuperscript{11}The details of the experimental set-up originally appeared in the paper (Andrighetto et al., 2008) for which a copy cannot be found on the Internet.
agent to infer norms). Agents infer norms when they join new societies and deliberate about norms. Agents can also suggest a new norm based on their past experience and may bring about norm change.

### 3.2.5 Cultural and evolutionary mechanisms

Researchers have proposed cultural and evolutionary models (Boyd and Richerson, 1985; Chalub et al., 2006) for norm spreading and emergence. Boyd and Richerson (1985) proposed that norms can be propagated through cultural transmission. According to them, there are three ways by which a social norm can be propagated from one member of the society to another. They are

- **Vertical transmission** (from parents to offspring)
- **Oblique transmission** (from a leader of a society to the followers)
- **Horizontal transmission** (from peer to peer interactions).

Of these three kinds of norm transmission mechanisms, vertical and oblique transmissions can be thought of as leadership mechanisms in which a powerful superior convinces the followers to adopt a norm. Horizontal transmission is a peer-to-peer mechanism where agents learn from day-to-day interactions from other peers. Few researchers have used this idea to experiment with norm spreading (e.g. the idea of normative advice used by Verhagen (2000)).

Norm spreading based on evolution involves producing offspring that inherit the behaviour of their parents. One well known work in this category is by Axelrod (1986). Other researchers have also experimented with evolutionary models for norm spreading (Chalub et al., 2006; Villatoro and Sabater-Mir, 2009).

Chalub et al. (2006) studied how norms might spread in different societies (e.g. an archipelago of islands). In their experiments, agents in an island were fully connected to each other. Each agent played the donor-receiver game once with all other agents in the island. Then an agent reproduced by choosing a connected agent at random and comparing their payoffs. If its payoff was higher than the other agent’s, then the other agent inherited
the strategy of the winning player. Each island had a Gross Domestic Product (GDP) (i.e. the score of the island was represented as the GDP) which was a normalised average payoff of the entire island population at end of playing the game. Islands which were fully connected competed against each other. There were times of war and peace. During peace, the norms of the islands did not change. When the islands were at war, they played the Hawk and Dove game (Smith and Price, 1973). The losers changed their norm based on a probabilistic norm-update rule. This model resulted in the emergence of one common social norm which induces reputation-based cooperative behaviour in these islands.

Discussion - Even though cultural models provide an answer to how norms are spread (e.g. normative advice from parents to children), they do not explain how a norm is internalized in the first place. There is an implicit assumption that the norms are internalized by the parents and are passed on to their progeny. A limitation of evolutionary models is that they do not distinguish a strategy from a norm. In other words, these models lack the conceptualization of normative expectation. Some researchers have studied the effect of gene-cultural co-evolution (Gintis, 2001). Kendal et al. (2006) have studied the cultural co-evolution of norm adoption and enforcement when punishers are rewarded or non-punishers are punished. Still, the issues that are not addressed remain the same. Chalub et al. (2006) note that they believe it is important to investigate how the cognitive capacity of an agent impacts the evolution of social norms as the cognitive capacity of humans allows them to set and enforce norms. Work along these lines is in progress (see works of Andrighetto et al. (2007; 2010a) and chapters 6-9 of this thesis).

3.2.6 Sanction mechanisms

Even though the models discussed in Section 3.2.2 are based on the notion of leadership, they do not include the ability to sanction agents that do not follow the norm specified by a norm leader. Several works on norms have used the notion of social power to inflict sanctions on agents that do not follow a norm (Axelrod, 1986; López y López et al., 2002; Flentge et al., 2001).

In his well known work, Axelrod (1986) has shown how a meta-norm of punishing an...
agent who did not enforce a norm can stabilise the norm of cooperation in an agent society. When an agent A sees another agent B violating a norm, then the agent A observing this violation can either punish or ignore this violation. If another agent C sees A not punishing B, it can choose to punish B with a certain probability. When this mechanism was introduced, the norm stabilised in every run of the experiment. The norm of cooperation was not always sustained without the use of a meta-norm mechanism. This shows that a meta-norm mechanism is useful for sustaining a norm. Axelrod also notes that when the cost of punishments are low for the punishers a norm can be sustained. This has been verified by other researchers (e.g. Flentge et al. (2001)).

López et al. (2002; 2003) have used the motivation of an agent and autonomy to be the main driving forces behind norm compliance and have considered punishments and rewards in their model. In their scheme agents have goals and norms. Their framework models agents with different personalities (social, pressured, opportunistic, greedy, fearful, rebellious, simple imitation, reasoned imitation and reciprocity). The model assumes that sanctions or rewards can only be used as an enforcing mechanism if they affect an agent's goals. The authors have shown the effect of norm compliance by varying different types of agents through simulations based on two parameters, the social satisfaction and individual satisfaction which respectively indicate the significance of norm compliance from the points of view of the society as well as an individual agent. One limitation of this work is that the cost of sanctions is not explicitly considered on the part of the agent imposing this punishment. They assume that all punishments and rewards are applied by someone else in a society.

Flentge et al. (2001) have shown how an agent comes to acquire a possession norm. In this system an agent can possess a piece of land. The authors have shown that the probability for the survival of the population is much higher when possession claims of others are respected. In the short term, an agent can benefit from violating the possession norm. They have noted that sanctions help in the establishment of the possession norm when the sanctioning costs are low or when there is no cost for sanctioning.

**Discussion** - The role of sanctions is to help sustain norms. In centralised institutional mechanisms there are special *norm policemen* who ensure that norm violations do not take place (e.g. governor agents in the AMELI framework (Arcos et al., 2005a)), and the insti-
tution bears the cost of enforcement. However, in a distributed punishment environment as employed in open agent societies, except for altruistic agents, the cost of sanctions may be high for the punishing agents. Altruistic agents have their own utility function that is different from the selfish agents (e.g. an altruistic agent’s utility increases when it performs a good deed). Even though the work in this area has begun (Ohtsuki et al., 2009; Fehr and Fischbacher, 2004), a proper account of the cost of punishment should be provided by future works that employ a sanction mechanism.

Another issue is that sanctions have an impact on an agent’s autonomy. Agents may have to give up some of their autonomy when they are forced to take up a norm in order to avoid the reduction of their utilities through sanctions. Researchers have modelled different personalities of agents to address this issue (e.g. social agents, Pressured agents) (López y López et al., 2002; Criado et al., 2010).

### 3.2.7 Reputation mechanisms

Reputation refers to the positive or negative opinion about a person or agent based on their interactions with others in the society. Researchers have addressed how reputation models are beneficial in sustaining norms in an agent society (Castelfranchi et al., 1998; Hales, 2002). They have experimented with the effect of a notion of “normative reputation” on the compliance costs of the norm. They have shown that providing a mechanism to record the normative reputation of agents of the society helps in redistributing the costs of norm compliance to both the agents that follow the norms as well as those who do not follow the norms.

The context of interaction of agents in the work of Castelfranchi et al. (1998) was a food-consumption problem where the food source randomly appears in a grid and the agent closest to it can either see or smell the food. Strategic agents were compared to normative agents. Strategic agents would attack another agent consuming food if that agent was weaker. Normative agents on the other hand were those that followed norms and in this case they followed the finders-keepers norm. Castelfranchi et al. (1998) showed that if the population had pure strategies (either completely strategic or normative), the normative agents always had a better utility at the end of the game as aggression costs both the aggressor and the
victim. For mixed strategies, the strategic agents always fared better than the normative agents because the strategic agents took advantage of the normative compliance of the agents. The researchers then introduced the notion of normative reputation where each agent learns whether another agent is normative or strategic. A normative agent shares this information with another normative agent. This helped the normative agents to identify the strategic agents. When the normative agents encountered a strategic agent, they did not follow the norm. In this way, they distributed the cost of norm compliance to the agents that did not observe the norm. Thus, the agents that used the normative reputation fared better in mixed strategy scenarios.

Hales (2002) has shown that group reputation supports the creation of beneficial norms. He extended the model described by Castelfranchi et al. (1998) to address the food-consumption game by stereotyping agents into homogeneous groups based on reputation. Agents belonged to one of two groups: the normative group or the strategic group (i.e. cheaters). The concept of group is similar to the tagging mechanism where each group can be thought of as having a common tag (i.e. each agent belongs to a certain tag group (Hales, 2001)). When an agent in the normative group interacts with an agent in the cheating group, the agent will sense that the opponent is a cheater. It then associates the cheating action with the group and communicates this information to all the other agents in the group. When other norm-following agents interact with agents in the cheating group, they will use the cheating strategy instead of the normative strategy. By this process, the agents are able to distribute the cost of norm compliance to the cheaters. The weakness of this approach is that even if one of the agents in the norm-following group becomes a black sheep, then the whole society is labelled as a cheating society because this model allows one agent’s misbehaviour to tarnish the reputation of the whole society. Also, it does not allow a group to revert back if it was stereotyped as a cheating group.

Discussion - Agents using reputation models store the reputation of the agents that they interact with. This can be either an individual reputation or a group reputation. Other researchers in multi-agent systems field have used multi-level reputation through referrals (Yu and Singh, 2003). Both the models discussed above do not take into account the personal cost incurred by an agent for maintaining the history of interactions with other agents that
is used for computing reputation. However, reputation is a simple decentralised mechanism and agents can deliberate on the cost of computing the reputation by varying the amount of history they would like to maintain.

### 3.2.8 Emotion-based mechanisms

Some researchers have modelled agents with emotions (Staller and Petta, 2001; Scheve et al., 2006; Fix et al., 2006). Based on the work done by Scheve et al. (2006), Fix et al. (2006) discuss the micro-macro linkage between emotions at the micro-level and norm enforcement at the macro-level. The authors argue that emotions have a norm-regulatory function in agent societies. An emotional agent that observes deviation of another agent from a norm might experience emotions such as contempt or disgust which can be the motivation behind sanctions. Agents that are sanctioned might experience emotions such as shame, guilt or embarrassment which might lead to norm internalization. The authors have used a Petri net model (Jensen, 1992) to capture the micro-macro linkage. It should be noted that the proposed model has not been implemented in the context of a simulation experiment. Staller and Petta (2001) extended the experimental set-up of Conte and Castelfranchi (1995) by including emotion-based strategies.

**Discussion** - Not many works in MAS have considered emotional agents since not all domains require agents to be emotional. Emotional agents are interface agents that interact with human users (e.g. a virtual patient avatar interacting with a doctor trainee). In emotion-based models for norms, it is assumed that the punisher and the norm-violator know what the norms of the society are. Only if they had known what the norms are, could any kind of emotion be generated. Initially, a norm could be a personal norm in the mind of one agent (e.g. a norm entrepreneur or a normative advisor). To start with, an agent might not know what the norms of a society are until it is punished a certain number of times for the deviant behaviour. It should be noted that emotion detection itself can be a complex problem, but this can be simplified if it can be assumed that these emotions are visible through external actions (e.g. an agent yells at a litterer in a park).
3.2.9 Research using network topologies

Social networks are important for norm spreading and emergence because in the real world, people are not related to each other by chance. They are related to each other through the social groups that they are in, such as work groups, church groups, ethnic groups and hobby groups. Information tends to percolate among the members of the group through interactions. Also, people seek advice from a close group of friends and hence information gets transmitted between the members of the social network.

In most empirical works, the treatment of norms has been mostly in the context of an agent society where the agents interact with all the other agents in the society in a random fashion (Boman, 1999; Verhagen, 2000). Few researchers have considered the actual topologies of the social network for norm emergence (Pujol, 2006). We believe such an approach is important for the study of norm spreading and emergence, as networks provide the topology and the infrastructure on which the norms can be exchanged. Researchers have studied different kinds of network topologies and their applications in the real world (an overview of different topologies is given by Mitchell (2006)). These application areas include opinion dynamics (Fortunato et al., 2005) and the spread of diseases (Cohen et al., 2003). Researchers in normative multi-agent systems have started to look at the role of network topologies (Kittock, 1995; Pujol, 2006; Villatoro and Sabater-Mir, 2009; Sen and Sen, 2010). Network topologies have also been explored by other multi-agent system researchers in other contexts, such as reputation management (Pujol et al., 2002; Yu and Singh, 2003).

Research that has considered network topologies can be categorised into static and dynamic network topology approaches. In the static approach, the network topology is fixed. In the dynamic topology approach, the underlying network can change when the simulation experiments are conducted.

3.2.9.1 Research using a static network topology

Kittock was the first to experiment on the role of network topology in convention emergence (Kittock, 1995). Agents interacted with each other based on their position in a circular lattice. The agents learned about the best strategy to choose based on the HCR algorithm proposed by Shoham and Tennenholtz (1997). Kittock noted that the choice of the global
structure has a profound effect on the evolution of the system. Depending upon the topology of the network, the emergence of a convention varies. In particular, he conjectured that the diameter of a network is directly related to the rate of convergence, which was observed by other researchers later (Barabasi and Albert, 1999; Albert and Barabasi, 2002).

Nakamaru and Levin (2004) studied how two related norms evolved in networked environments. The two related norms were based on a) the background of an agent (e.g. the religion that the agent follows) and b) the opinions that the agent holds (e.g. opinions about fast food, political affinity). The background is a norm that the whole population shares but different opinions can be held by agents in that population. They note that when people of the same background meet, they might change some of their opinions and when all opinions of two agents are the same, they can change their background if they are different. They conducted experiments using four different types of network topologies and observed how the two related norms evolved on those networks. They showed that the spread of social norms is influenced not only by the structure of the network topology but also by the learning mechanism that is used. They experimented using mechanisms of social learning (imitation) and individual learning (learning based on interactions with other agents). They showed that imitation does not lead to co-existence of social norms (which has also been reported by Campenní et al. (2009)), while individual learning does. They also showed that individuals learn faster using imitation than when using individual learning. They observed that norms tend to propagate the fastest on a scale-free network.

Anghel et al. (2004) investigated the effects of inter-agent communication across a network in the context of playing the minority game (Challet and Zhang, 1998). They showed that a scale-free leadership structure emerges on top of a random network. Pujol (2006) dealt with the emergence of conventions on top of social structures. He used the HCR mechanism proposed by Shoham and Tennenholtz (1997) to test norm emergence in connected, random, small world and scale-free networks. He also demonstrated that the structure of the network is crucial for norm emergence. Recently several researchers (Salazar et al., 2008; Villatoro et al., 2009; Sen and Sen, 2010) have investigated the role of static network topologies on norm emergence in different domains.
3.2.9.2 Dynamic topology works

Very few researchers have investigated the role of dynamic network topologies on norm spreading and emergence. Griffiths and Luck (2010) have experimented on how norms might emerge on a network topology where an agent rewire its links by replacing its worst neighbours with the best neighbours of its immediate neighbours. Even though the work considers the links of a node to be rewired, the work does not model an open agent society where new agents can join and leave the society. Another limitation of this work is that the concept of tag is viewed as a norm (i.e. agents belonging to a tag group have the norm). This approach does not allow multiple norms to co-exist in a group.

3.2.9.3 Discussion

One limitation of static topologies is that they assume that the underlying network topology does not change. In the real world and in artificial agent societies, the social network topologies are dynamic. People join and leave social groups, so any mechanism for norm emergence should be applicable to dynamically changing network topologies. Thus, models for norm spreading and emergence should be tested on top of dynamically changing network topologies. Another limitation is that the weights of the links are not considered. The strength of the links from an agent to other agents might be different which may influence norm spreading and emergence. Autonomy of the agents is also not considered in most of these works. Some agents can just refuse to participate in norm spreading. If they are the important nodes such as a hub, this will have implications on the spread of the norm. In most of the models, individuals belong to only one group but in reality the agents can be influenced from many angles as an individual might belong to several groups (e.g. work group, neighbours and hobby group). Each of these groups could have a different topology.

3.3 Characteristics of empirical works on norms

In this section we describe the important characteristics of individual agents and agent societies identified based on the empirical works on norms discussed in Section 3.2. Table 3.2 shows a comparison of the characteristics of an individual agent as studied in these research
works. Table 3.3 provides a comparison of the characteristics of an agent society.

### 3.3.1 Characteristics of an agent

The norm-related characteristics that have been identified are based on answering the questions: a) whether an agent can identify\(^{12}\), internalize, violate, influence\(^{13}\), represent and communicate norms and b) what the context of interaction between agents in the society is. The characteristics identified are discussed below.

#### 3.3.1.1 Norm identification

In many research works, a norm is considered a strategy (see Column 2 of Table 3.2). Many works consider a norm as an utilitarian strategy that maximizes the agent’s utility. The agents in these works do not possess the notion of “normative expectation”. For example, game-theory based works view a norm as a strategic action that leads to a high payoff (Shoham and Tennenholtz, 1992; Sen and Airiau, 2007) or a strategy that requires an agent to do the least amount of computation, thus maximizing the time it can spend on other activities (Epstein, 2001). Most works that view norm as the best strategy use machine learning approaches to find the best strategy (Shoham and Tennenholtz, 1992; Walker and Wooldridge, 1995; Sen and Airiau, 2007). A norm has also been considered as a strategy given to an agent by the designer during design time (offline strategy). Most of the works that consider norm as a strategy deal with conventions (i.e. behavioural regularity). Most of these works do not model the ability of agents to violate norms, hence, do not consider norm enforcement. However, a few researchers focus on how a norm is recognized by employing cognition-based deliberative processes. In other words they aim to address the question how an agent can identify a norm in an agent society at run-time (Andrighetto et al. (2010a) and Chapters 6-9 of this thesis).

---

\(^{12}\)In order to simplify the characteristics of the agents, the aspect of norm creation has been folded into norm identification.

\(^{13}\)We use the term influence to characterize the ability of an agent to spread and/or enforce norms.
<table>
<thead>
<tr>
<th>Model</th>
<th>Identification</th>
<th>Internalization</th>
<th>Violation</th>
<th>Influence</th>
<th>Representation</th>
<th>Communication</th>
<th>Interaction context</th>
</tr>
</thead>
<tbody>
<tr>
<td>Axelrod (1986)</td>
<td>Utilitarian strategy</td>
<td>No</td>
<td>Yes</td>
<td>Yes</td>
<td>Integers</td>
<td>Yes</td>
<td>Norms game</td>
</tr>
<tr>
<td>Shoham and Tennenholtz (1992)</td>
<td>Utilitarian strategy</td>
<td>No</td>
<td>No</td>
<td>No</td>
<td>Integer</td>
<td>No</td>
<td>Coordination game</td>
</tr>
<tr>
<td>Kittock (1995)</td>
<td>Utilitarian strategy</td>
<td>No</td>
<td>No</td>
<td>No</td>
<td>Integer</td>
<td>No</td>
<td>Coordination &amp; cooperation games</td>
</tr>
<tr>
<td>Conte and Castelfranchi (1995)</td>
<td>Offline strategy</td>
<td>No</td>
<td>No</td>
<td>No</td>
<td>Boolean</td>
<td>No</td>
<td>Food searching game</td>
</tr>
<tr>
<td>Walker and Wooldridge (1995)</td>
<td>Offline strategy</td>
<td>No</td>
<td>No</td>
<td>No</td>
<td>Boolean</td>
<td>Yes</td>
<td>Food searching game</td>
</tr>
<tr>
<td>Castelfranchi et al. (1998)</td>
<td>Offline strategy</td>
<td>No</td>
<td>No</td>
<td>No</td>
<td>Boolean</td>
<td>Yes</td>
<td>Food searching game</td>
</tr>
<tr>
<td>Verhagen (2001)</td>
<td>Utilitarian strategy</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>Decision tree</td>
<td>Yes</td>
<td>Resource consumption game</td>
</tr>
<tr>
<td>Epstein (2001)</td>
<td>Utilitarian strategy</td>
<td>No</td>
<td>No</td>
<td>No</td>
<td>Binary</td>
<td>No</td>
<td>Coordination game</td>
</tr>
</tbody>
</table>

*Note: Yes - considered, No - not considered, NA - Not Applicable, NS - Not Specified*
<table>
<thead>
<tr>
<th>Model</th>
<th>Identification</th>
<th>Internalization</th>
<th>Violation</th>
<th>Influence</th>
<th>Representation</th>
<th>Communication</th>
<th>Interaction context</th>
</tr>
</thead>
<tbody>
<tr>
<td>Hales (2002)</td>
<td>Offline strategy</td>
<td>No</td>
<td>No</td>
<td>Yes</td>
<td>Boolean</td>
<td>Yes</td>
<td>Food searching game</td>
</tr>
<tr>
<td>Hoffmann (2003)</td>
<td>Utilitarian strategy</td>
<td>No</td>
<td>No</td>
<td>Yes</td>
<td>Integer</td>
<td>No</td>
<td>Pick a number game</td>
</tr>
<tr>
<td>López y López (2003)</td>
<td>Offline strategy</td>
<td>No</td>
<td>Yes</td>
<td>Yes</td>
<td>NA</td>
<td>No</td>
<td>NS</td>
</tr>
<tr>
<td>Nakamaru and Levin (2004)</td>
<td>Offline strategy</td>
<td>No</td>
<td>No</td>
<td>No</td>
<td>Binary string</td>
<td>No</td>
<td>Background and opinion exchange</td>
</tr>
<tr>
<td>Chalub et al. (2006)</td>
<td>Utilitarian strategy</td>
<td>No</td>
<td>No</td>
<td>No</td>
<td>8-bit string</td>
<td>No</td>
<td>Donor-Receiver, Hawk-Dove games</td>
</tr>
<tr>
<td>Pujol (2006)</td>
<td>Utilitarian strategy</td>
<td>No</td>
<td>No</td>
<td>No</td>
<td>Binary</td>
<td>No</td>
<td>Coordination game</td>
</tr>
<tr>
<td>Sen and Airiau (2007)</td>
<td>Utilitarian strategy</td>
<td>No</td>
<td>No</td>
<td>No</td>
<td>Binary</td>
<td>No</td>
<td>Coordination game</td>
</tr>
<tr>
<td>Andrighetto et al. (2010a)</td>
<td>Cognition</td>
<td>No</td>
<td>No</td>
<td>Yes</td>
<td>Boolean</td>
<td>Yes</td>
<td>NS</td>
</tr>
<tr>
<td>Chapters 4 and 5</td>
<td>Utilitarian strategy</td>
<td>No</td>
<td>No</td>
<td>Yes</td>
<td>Integer</td>
<td>Yes</td>
<td>Ultimatum game</td>
</tr>
<tr>
<td>Chapters 6-9</td>
<td>Cognition</td>
<td>No</td>
<td>Yes</td>
<td>Yes</td>
<td>String</td>
<td>Yes</td>
<td>Park and restaurant scenarios</td>
</tr>
</tbody>
</table>

Note: Yes - considered, No - not considered, NA - Not Applicable, NS - Not Specified
3.3.1.2 Norm internalization

In human societies, people internalize norms. Internalization refers to the process by which an agent makes the norm its own (making a group norm as its own personal norm) and the norm conformity on the part of the agent does not require external punishments or rewards (Aronfreed, 1968; Andrichetto et al., 2010b). Modelling the concept of internalization in a software agent has not been dealt with by many researchers (see Column 3 of Table 3.2). Verhagen (2001) has studied norm internalization from a computational viewpoint. In his model, the difference between an agent’s personal view of a norm and the group view of the norm is used to measure norm internalization. Recently, a cognition-based model for norm internalization has been proposed by Conte et al. (2010).

3.3.1.3 Norm violation

Researchers note that behavioural autonomy with respect to norms (also called norm autonomy) is an important attribute of a normative agent (Dignum, 1999; Verhagen, 2000). An agent participating in a normative system should have the ability to say no and violate a norm when it does not want to obey the norms (because the norm may prevent the satisfaction of an urgent goal or may lower the agent’s utility drastically). However, not many normative agent models have considered the autonomy of agents (see Column 4 of Table 3.2). Additionally, the notion of adjustable autonomy that relates to norms has been discussed by Verhagen and Boman (1999). The agent should have the ability to dynamically change the degree to which it follows a norm. This creates a dilemma on the part of an agent whether to give up some of its autonomy in order to be a part of a group or to achieve a common goal. Still, the agents that are a part of a group that obey certain norms should be able to identify whether a norm applies in certain situations. For example, when the traffic light turns red at 3am and the agent does not see any other agent in an intersection, it should be able to decide whether to respect the norm. We believe, the notion of dynamic autonomy with reference to norms can be considered in the future works on norms.
3.3.1.4 Norm influence

An agent can influence another agent’s norm either through persuasion or coercion. A leader can persuade his/her followers to follow a norm, thus contributing to the spread of a norm. An agent can use its social power to coerce (punish or reward) another agent to follow a norm (i.e. norm enforcement). Punishments may result in the loss of utility or the reputation of the punished agent. Punishments may trigger emotions in agents (e.g. an agent violating a norm it has internalized may feel remorse for having done so). It can be observed from Column 5 of Table 3.2 that only some of the research works have considered the ability of agents to influence the norms of the other agents.

3.3.1.5 Norm representation

Norm representation is the ability of an agent to explicitly represent the norm in some form. Even though norm representation is studied widely in NorMAS (see Section 2.3.5), norm representations used in the simulation models are all based on simple data types (see column 6 of Table 3.2). This is mainly because keeping a representation simple reduces the complexity of the model and reduces the computation required. Additionally it allows exploration of certain properties which may not be easily understood when a system has a high level of complexity.

In Axelrod’s work (1986), the norm against cheating was inferred based on the boldness and vengefulness parameters possessed by the agent. These parameters were integer values. Though the norm representation was not explicitly stated by Axelrod, it can be inferred by the reader. In Table 3.2, whenever the reader has to infer what the norm representation might have been, we have used italicised text (see column 5). In some research, a norm is a binary variable, e.g. to indicate whether an agent drives on the left or right (Sen and Airiau, 2007; Epstein, 2001). Castelfranchi et al. (1998) used a Boolean variable to represent a norm. Chalub et al. (2006) used a string representation for norms.

In complex, dynamic and open agent societies, these representations may not be sufficient. For example, in the context of park littering, agents observe what kinds of norms might evolve in the usage of a park. It could happen that littering is prohibited in general. One kind of norm could be that whenever an agent litters within 20 metres from a rubbish bin, it would
be punished. It could also be that whenever there is no one within ten metres of the agent’s vicinity and no rubbish bin could be found within 20 metres of the agent then littering would be permitted. In these cases, the representation of the norm needs to be flexible and dynamic, and for this reason a richer model of norm representation would be required. This dynamic model of norm representation should allow for norms to change at run-time. It should also allow more subtle contextual norms to emerge. In this thesis norms are considered as simple propositions (e.g. \textit{prohibit(litter)}) in Chapters 7 and 8. Simple representations of conditional norms have been used in Chapter 9.

3.3.1.6 Norm communication

Agents should also have the ability to communicate norms to other agents. Some research works have considered explicit norm communication where agents send normative advice to other agents when asked for advice (Verhagen (2000), chapters 4 and 5 of this thesis). Research employing game-theoretical models assumes that an agent explicitly knows the setting and options available to other agents (i.e. the payoff matrix). In these works, there is no explicit norm communication. Hence, there is no need for explicit representation of norms. Only few works have considered the notion of norm communication (Walker and Wooldridge, 1995; Hales, 2002) (see column 7 of Table 3.2).

It should be noted that if the state space of actions performed by the agents increases, norm communication becomes important as agents may not know the actions that are prohibited or obliged \textit{a priori} at design time. Additionally, norm communication will be beneficial to newly joining agents as well as other agents in the society. Newly joining agents can ask other agents about the norms in the society, and the existing agents can use communication channels to enquire if a norm is currently in force. It should be noted that communication of norms will require an agent to have an explicit representation of norms.

3.3.1.7 Interaction context

Researchers have considered simple scenarios for their simulation experiments with norms (see column 8 of Table 3.2). These scenarios mainly include food sharing game (discussed in Section 3.2.7), and the coordination and cooperation games. This thesis makes use of
the ultimatum game (Slembeck, 1999) for the works on norm emergence (chapters 4 and 5) and makes use of two scenarios borrowed from human societies which are the park and the restaurant scenarios, in the context of norm identification (chapters 6-9).

3.3.2 Characteristics of an agent society

Based on the study of different empirical works on norms (Section 3.2), we have identified important characteristics of an agent society. These include, a) the nature of the society that is being modelled (Fixed vs. Dynamic) and the size of the society, b) the consideration of network topology, c) the consideration of norm coexistence, d) the consideration of norm conflict, and e) the norm emergence criteria used.

3.3.2.1 Nature of the society and its size

Even though most researchers investigating empirical works on norms point to open systems as their target environment, their works have used only a fixed number of agents (see column 2 of Table 3.3). The number of agents used in their simulations do not change at run-time. Under an open world assumption, agents should be able to join and leave dynamically. Sen and Airiau (2007) have experimented with adding agents dynamically to a society. We have considered agents living for certain amount of time and then being replaced with new agents (Chapter 5) and also have modelled agents that are able to move from one society to another (chapters 7 and 8).
Table 3.3: Comparison of characteristics of an agent society in the simulation works on norms

<table>
<thead>
<tr>
<th>Model</th>
<th>Nature of the agent society and (size)</th>
<th>Network topology</th>
<th>Norm coexistence</th>
<th>Norm change</th>
<th>Norm conflict</th>
<th>Emergence criteria</th>
</tr>
</thead>
<tbody>
<tr>
<td>Axelrod (1986)</td>
<td>Fixed (20)</td>
<td>Random</td>
<td>No</td>
<td>No</td>
<td>No</td>
<td>100%</td>
</tr>
<tr>
<td>Shoham and Tennenholtz (1992)</td>
<td>Fixed (100)</td>
<td>Random</td>
<td>No</td>
<td>No</td>
<td>No</td>
<td>85-95%</td>
</tr>
<tr>
<td>Kittock (1995)</td>
<td>Fixed (10-10000)</td>
<td>Circular lattice</td>
<td>No</td>
<td>No</td>
<td>No</td>
<td>90%</td>
</tr>
<tr>
<td>Conte and Castelfranchi (1995)</td>
<td>Fixed (50)</td>
<td>Random</td>
<td>No</td>
<td>No</td>
<td>No</td>
<td>NA</td>
</tr>
<tr>
<td>Walker and Wooldridge (1995)</td>
<td>Fixed (100)</td>
<td>Random</td>
<td>No</td>
<td>No</td>
<td>No</td>
<td>73-99%</td>
</tr>
<tr>
<td>Castelfranchi et al. (1998)</td>
<td>Fixed (50)</td>
<td>Random</td>
<td>No</td>
<td>No</td>
<td>No</td>
<td>NA</td>
</tr>
<tr>
<td>Verhagen (2001)</td>
<td>NS</td>
<td>Fully-connected</td>
<td>No</td>
<td>No</td>
<td>No</td>
<td>NS</td>
</tr>
<tr>
<td>Epstein (2001)</td>
<td>Fixed (191)</td>
<td>Ring</td>
<td>Yes</td>
<td>No</td>
<td>No</td>
<td>Varies</td>
</tr>
<tr>
<td>Hales (2002)</td>
<td>Fixed (50)</td>
<td>Random</td>
<td>No</td>
<td>No</td>
<td>No</td>
<td>NA</td>
</tr>
<tr>
<td>Hoffmann (2003)</td>
<td>Fixed (10-50)</td>
<td>Fully-connected</td>
<td>No</td>
<td>No</td>
<td>No</td>
<td>70%</td>
</tr>
<tr>
<td>López y López (2003)</td>
<td>NS</td>
<td>NS</td>
<td>No</td>
<td>No</td>
<td>No</td>
<td>NS</td>
</tr>
</tbody>
</table>

Note: Yes - considered, No - not considered, NA - Not Applicable, NS - Not Specified
Continued on next page
<table>
<thead>
<tr>
<th>Model</th>
<th>Nature of the agent society and (size)</th>
<th>Network topology</th>
<th>Norm coexistence</th>
<th>Norm change</th>
<th>Norm conflict</th>
<th>Emergence criteria</th>
</tr>
</thead>
<tbody>
<tr>
<td>Nakamaru and Levin (2004)</td>
<td>Fixed (900-10000)</td>
<td>4 types of networks</td>
<td>No</td>
<td>No</td>
<td>No</td>
<td>Varies</td>
</tr>
<tr>
<td>Chalub et al. (2006)</td>
<td>Fixed (128)</td>
<td>Fully-connected</td>
<td>Yes</td>
<td>No</td>
<td>No</td>
<td>95-98%</td>
</tr>
<tr>
<td>Pujol (2006)</td>
<td>Fixed (Upto 100000)</td>
<td>4 types of networks</td>
<td>No</td>
<td>No</td>
<td>No</td>
<td>90%</td>
</tr>
<tr>
<td>Sen and Airiau (2007)</td>
<td>Fixed (100) and dynamic</td>
<td>Random</td>
<td>Yes</td>
<td>No</td>
<td>No</td>
<td>Mostly 100%</td>
</tr>
<tr>
<td>Andrighetto et al. (2010a)</td>
<td>Fixed (100)</td>
<td>Random</td>
<td>Yes</td>
<td>No</td>
<td>No</td>
<td>NS</td>
</tr>
<tr>
<td>Chapters 4 and 5</td>
<td>Fixed (100) and dynamic</td>
<td>3 types of networks</td>
<td>Yes</td>
<td>Yes</td>
<td>No</td>
<td>Varies (35-100)%</td>
</tr>
<tr>
<td>Chapters 6-9</td>
<td>Fixed (50-100) and dynamic</td>
<td>Random</td>
<td>Yes</td>
<td>Yes</td>
<td>No</td>
<td>Varies</td>
</tr>
</tbody>
</table>

Note: Yes - considered, No - not considered, NA - Not Applicable, NS - Not Specified

3.3.2.2 Consideration of network topology

Agents form links with other agents that they interact with. In the context of norms these links are important, because normative influence spreads through these links. While the emphasis is on static network topologies (such as the work by Kittock (1995) and Pujol
(2006)) not many researchers have considered the role of dynamic network topologies and the effect of norm spreading and emergence (see column 3 of Table 3.3). Chapter 5 of this thesis makes some contributions to the construction of dynamic network topologies.

3.3.2.3 Consideration of norm coexistence

An area where little research has so far been conducted is the consideration of societies where different norms co-exist in a given context (see column 4 of Table 3.3). Most works have concentrated on how one particular norm emerges in a society. Chalub et al. (2006) investigated how a norm might emerge when agents living in different societies (islands) with different norms are brought together after war. We believe more research in this area can be undertaken. Especially considering the richness of virtual agent societies with different norms for the same context, further extensions can be made, such as investigating when a norm will take over and dominate another when two societies interact and when these norms can co-exist. In chapter 5 of this thesis we have studied how different societies with different norms for the same context that are brought together due to environmental factors might arrive at a common norm.

3.3.2.4 Consideration of norm change

One issue that has not been addressed by empirical works on norms is the dynamic change of norms. None of the works that are based on simulations address how a norm is made obsolete or how a new norm replaces an old one. Similar to human societies, agent societies should allow norms to change or evolve (either based on change in the environment or due to finding a superior norm). Normative simulation systems that cater for the dynamic change of norms will be useful for understanding the conditions under which new norms replace the older ones, both in human and artificial agent societies. Norm change in agent societies is discussed in Chapters 7 and 8 of this thesis.

14This has been studied by deontic logic researchers in the context of an open institution where norms are added and removed (Vasconcelos et al., 2009).
3.3.2.5 Consideration of norm conflicts

None of the empirical works on norms have modelled the ability of agent societies to address norm conflicts (see column 5 of Table 3.3). However, using a formal approach, some researchers have worked on resolving normative conflicts in open agent societies (Vasconcelos et al., 2009; Kollingbaum et al., 2008). Future empirical models of norms can address how to simulate, identify, and resolve normative conflicts that have richer and more complex constructs.

3.3.2.6 Norm emergence criteria

Finnemore and Sikkink (1998) note that when a critical mass of agents has accepted a new idea (a norm) as appropriate, then a norm is said to have emerged. It should be noted that researchers have used different criteria for norm emergence (see column 5 of Table 3.3). The criteria varies from 35% to 100%.

3.3.2.7 Other characteristics

An important characteristic of an open agent society is that the new agents that join the society can have different personality types (e.g. social agents (agents that always follow norms), rebellious agents (agents that always disobey norms) and opportunistic agents (agents that violate norms opportunistically)). Hence the composition of agents in a society will play an important role in norm dynamics. Simulation models need to cater for this kind of dynamism in the society. The work by López y López (2003) is one such approach in this direction (varying the personality of the agents). We believe experimenting with dynamic composition of agents in open agent societies is a good avenue for further research. This thesis considers some personality types of agents based on simple roles (chapters 4-9).

Another attribute that has not been considered by many researchers is the consideration of a noisy environment. A noisy environment is an environment where the information that an agent receives from other agents or the environment may have been distorted because of the noise (e.g. communication noise can distort the quality of the information received by the agent). Noise is modeled as a parameter (e.g. 10% noise in the system) which represents the amount of distortion in the information that is communicated to another agent. It should
be noted that only Hoffmann (2003) has considered the possibility of a noisy environment. We believe, similar to his approach, noise could be easily modeled as a parameter in future empirical works on norms.

### 3.3.3 Discussion

In this section we have described the important characteristics of an individual agent and an agent society that have been identified based on the research works on norms discussed in Section 3.2. We have also compared the identified characteristics across different research works. We have also discussed the relevant characteristics considered by this thesis.

### 3.4 Contextualizing the contributions of this thesis

This section contextualizes the empirical contributions of this thesis based on the categorization presented in Section 3.2. A full summary of the contributions of the thesis is provided in Chapter 10.

The two main contributions of this thesis are to the areas of norm emergence and norm identification. Emergence of norms can be observed from a bird’s-eye view of the entire society, while identification of norms is from the perspective of a single agent. Figure 3.4 shows a schematic representation of these two views.

An external agent (or the system agent shown at the top of the figure) that views the society from a bird’s eye view observes the dynamics of the spread of a norm (i.e. how a norm that has been created spreads in the society). The agent can observe the extent of spread of the norm at any point in time. It does not know the operational details of the individual agents with respect to the norm. For example, it does not know how an agent may recognize the norm. On the other hand, individual agents may employ different processes to recognize norms (e.g. machine learning, imitation) and upon recognition they may or may not follow the norm. Additionally, an agent that appears to be following a norm may not have internalized the norm. A norm-abiding agent may or may not participate in norm spreading (e.g. through persuasion or coercion). These two perspectives of norms (bird’s-eye view and an individual agent’s view) have been considered by this thesis. The focus of chapters 4 and
5 are on norm emergence from a bird’s-eye view. The focus of chapters 6 to 9 is on norm identification and the viewpoint is of an individual agent. The experimental results presented in chapters 4 to 9 are based on simulating artificial agents that are proxies for human agents.

The following sub-sections explain these two views and also contextualize the contributions of this thesis by providing pointers to the mechanisms that have been presented in Section 3.2.
3.4.1 Norm emergence

Figure 3.4 shows an agent society with several agents. Agents may play different roles in a society. The colour of the agent is an abstract representation the role it plays. Agents interact with other agents they are connected to in some context (e.g. work group). These interactions are shown as arrows between agents. Norms might emerge based on these interactions. An external observer (e.g. a system agent) can detect the waxing and waning of norms in an agent society (shown at the top of the figure).

The contributions of the thesis on norm emergence can be mapped to two categories of empirical works on norms in Figure 3.5 (highlighted in blue text). The contributions to norm emergence are: a) the proposal and the implementation of a distributed leadership mechanism for norm emergence (Chapter 4 and 5) and b) the consideration of dynamic network topology (Chapter 5).

3.4.1.1 Distributed leadership mechanism based on role-models

Centralized mechanisms for norm spreading and emergence have been investigated by researchers (Boman, 1999; Verhagen, 2001). We have proposed and experimented with a distributed approach for norm emergence based on role-models. In this approach, there could be several normative advisors (called role models) from whom other agents can request advice. In this model, an agent can be a leader for some agents and also it can be a follower of some other agent at the same time. An agent in this model chooses a role model based on the performance of the agents in its neighbourhood. The role model can recommend its norm to its followers who ask for advice. This approach allows for a hierarchy of leaders to emerge. Additionally, depending upon a leader’s neighbourhood, this approach may allow different norms to appear in each of the neighbourhoods. The mechanism has been tested on three types of network topology structures. The details of this mechanism are explained in Chapter 4. The contribution of this chapter falls under the leadership & entrepreneurship mechanisms and static network topology categories, as shown in Figure 3.5.
3.4.1.2 Creating and experimenting with dynamic network topologies

In order to model agents joining and leaving the society dynamically, we used a model of Gonzalez et al. (2006b) to create dynamic network topologies. Gonzalez et al. developed a model for constructing dynamically changing networks using the concept of colliding particles which represent agents interacting in an abstract social space. We have created dynamic network topologies using this model to test their agent-based role model leadership mechanism. We have also extended their model to create scenarios which depict how different societies can be brought together. We show how different types of norms emerge when societies with different norms for the same context (playing the Ultimatum game (Slembeck,
are brought together. In particular, we show that under certain conditions norms can co-exist in an agent society. The details of this mechanism is explained in Chapter 5. The contribution of this chapter falls under the dynamic network topology category.

### 3.4.2 Norm identification

An agent operating in an open agent society should have the ability to identify new norms, detect changing norms, and remove norms that do not hold. For this purpose, the agent should possess capabilities of inference based on the observing its local environment. The local environment (horizon) of an agent is given by a dashed square in Figure 3.4. The agent in the centre of the square can interact with four other agents that are present in its environment. The yellow “thought” bubble that appears on top of an agent represents the process employed by an agent for norm identification.

The two contributions to norm identification are: a) the proposal and the experimentation of an architecture for a cognitive agent to infer norms based on observations of interactions between agents (Chapter 6 to 9) and b) the use of association-rule mining, a data mining approach to infer norms (Chapters 6 to 9).

Towards answering the question of how agents identify a norm in open agent societies, we have proposed an internal architecture of a cognitive agent (Chapter 6) to identify a norm using a bottom-up approach. An agent can use this architecture to identify prohibition and obligation norms (Chapters 7 and 8, respectively). The norm inference component of the architecture makes use of association-rule mining, a data mining approach to infer norms.

We note that work related to our work in the area of norm identification is being carried out by the researchers involved in the EMIL project (Andrighetto et al., 2007). Researchers involved in the EMIL project are working on a cognitive architecture for norm emergence (for a description see Section 3.2.4). Our work differs from this work in four ways. Firstly, in our architecture we have chosen “reaction” or “signalling” (positive and negative) to be a top-level construct for identifying potential norms when the norm of a society is being shaped. We note that a sanction not only may imply a monetary punishment, it could also be an action that could invoke emotions (such as an agent yelling at another might invoke shame or embarrassment on another agent), which can help in norm spreading. Agents can
recognize such actions based on their previous experiences. Secondly, based on association-rule mining (Ceglar and Roddick, 2006), we propose algorithms for norm inference, which can be adapted by an autonomous agent for flexible norm identification. To the best of our knowledge, we are one of the first to use a data-mining approach to infer norms. Thirdly, we identify two different sets of norms in an agent’s mind: candidate norms and identified norms. Fourthly, our approach can be used to identify conditional norms (Chapter 9) and also it can be potentially used to detect conflicting norms (discussed in Chapter 10).

While many well-known works identify one norm that exists in the society (Hoffmann, 2003; Epstein, 2001; Axelrod, 1986), the internal agent architecture proposed in this thesis (Chapter 6) is able to identify several norms that might exist concurrently in the society. This work also addresses how an agent might be able to identify whether a norm is changing in a society and how it might react to this situation. In this model, the agents are able to identify the norm, as well as change and dynamically add, remove and modify norms.

The contributions of the norm identification aspect of this thesis can be mapped mainly to two categories which are cognitive mechanism and learning mechanisms (see Figure 3.5). We have contributed to the new sub-category of data mining under learning mechanisms for identifying norms. Since sanctions are used as a starting point for identifying norms, the thesis also contributes to the sanction mechanism for norm enforcement.

3.5 Summary

This chapter identified and described the five phases of a life-cycle model for norms based on the empirical works on norms. It categorized empirical works on norms into nine main categories and discussed the key papers in each of the categories. Based on the empirical models of norms, important characteristics of individual agents and agent societies were identified. These identified characteristics were then used to compare different research works. Finally, the contributions of this thesis were explained in the context of the categorization proposed in this chapter.
Chapter 4

Role model based mechanism for norm emergence

4.1 Introduction

In human societies, individuals are influenced by the ideas and opinions of others in the society. The transmission of cultural artifacts such as ideas, opinions and norms from one individual to another has been studied by several researchers (Cavalli-Sforza and Feldman, 1981; Boyd and Richerson, 1985). An individual may be able to influence the cultural artifacts of others (Boyd and Richerson, 1985). Boyd and Richerson (1985) have proposed that norms in human societies are spread from parents to children (i.e. vertical transmission), from a leader of a society to the followers (i.e. oblique transmission) and through peer to peer interactions (i.e. horizontal transmission).

In a social setting, as agents interact with each other, a social network structure is created. These social structures govern how agents interact with each other. Social network structures play an important role in governing how information is spread in a society (Mitchell, 2006). In real world, people are related to each other through the social groups that they are in, such as work groups, church groups, ethnic groups or hobby groups. People seek advice from a close group of friends and hence information gets transmitted between the members of the social network. Therefore, it is important to study norm emergence on top of social networks with realistic network topologies. Recently, this area of research has received
attention among multi-agent researchers (Pujol, 2006; Sen and Airiau, 2007; Villatoro and Sabater-Mir, 2009; de Pinninck et al., 2008; Salazar et al., 2009; Villatoro et al., 2009; Zhang and Leezer, 2009). The role of network topology has been studied in detail in other contexts such as opinion dynamics (Fortunato, 2005) and the spread of diseases (Cohen et al., 2003).

In this chapter we discuss a mechanism for norm spreading and emergence based on role models for software agents. The mechanism uses the concept of normative advice whereby the role models provide advice to the follower agents. Our mechanism is built using two layers of networks, the social link layer and the leadership layer. The social link layer represents how agents are connected to each other. The leadership layer represents the overlay network that is formed based on the role played by each agent. The two kinds of roles are leaders and followers.

This chapter is organized as follows. Section 4.2 provides an overview of the role model mechanism. Section 4.3 provides a background on social network topologies. Section 4.4 describes how the role model mechanism was tested on top of different network topologies and the context of interaction between agents. In Section 4.5 we describe the experimental set-up and also present our findings on how norms emerge on top of the leadership network when the topology of the social link network changes. The three kinds of social link networks that we have experimented with are fully connected networks, random networks, and scale-free networks. A discussion of the results is given in Section 4.6 and a summary is given in Section 4.7.

### 4.2 Role model mechanism

The Merriam-Webster dictionary (2010) defines a role model as a person whose behaviour in a particular role is imitated by others. The role models are agents who the societal members may wish to follow. The inspiration is derived from human society, where one might want to use successful people as a guide for one’s own actions. Any agent in the society can become a role model agent if some other agent asks for its advice. The role model agent represents a role model or an advisor who provides normative advise to those who ask for help. In our mechanism, each agent will have at most one role model.

An agent chooses a role model from its local neighbourhood (i.e. the agents it is directly
connected to) based on the performance of these neighbours. We assume that an agent knows the performances of all the agents it is connected to. This is based on the assumption that people who are successful in the neighbourhood are usually easily recognizable. We suggest that the success of an agent can be attributed to its norms\textsuperscript{1}. For example, an agent driving on the left can be considered successful in a society which drives on the left. An agent, by knowing the performance of other agents in the society, can choose the agent with the highest utility in its neighbourhood as its role model. The agent can then ask for normative advice from the role model. When the role model provides normative advice, the requester agent can choose to make use of the normative advice by adopting the norm recommended by the role model agent.

Agents are autonomous. The role model agent may refuse to provide normative advice in which case the requester agent can ask the next highest performing agent in its neighbourhood. After receiving the normative advice, an agent may not take the advice of the role model. An agent may gradually move towards the norm recommended by the normative advisor instead of moving instantly from one norm to another. For example, let us assume that a norm is an integer value (e.g. in the context of tipping waiters in a restaurant). If a requester agent has a norm to tip 5% of the bill and the role model agent has recommended a value of 15%, then the requester agent may move to a new value of 8% (which is based on a parameter that is set for each agent).

In our experiments, agents in a society are viewed as nodes and the associations between agents are viewed as links. Agents (nodes) and their interactions with other agents (links) together form a social network topology structure. The role model mechanism is tested on top this network topology. The role model mechanism for an agent is given in Algorithm 4.1. Each agent is allowed a certain number of interactions with its neighbours during which it plays the Ultimatum game (Slembeck, 1999) which is described in Section 4.4.1.1. At the end of interactions, the performance scores of the agents are computed. We also assume that the performance scores of agents are known within a given neighbourhood (e.g. an agent

\textsuperscript{1}We note that the success of an agent can be attributed to its own personal beliefs and values (a personal norm) which may eventually become a norm in the society (a group norm) through spreading. In this work, we attribute the success of an agent to its personal norm. This personal norm of the agent might be beneficial to the society if it becomes the group norm.
connected to five other agents will know the performance scores of these five agents). Then the agent contacts the other agents in its neighbourhood for normative advice based on the performance. An agent first contacts the best performer for advice and if that agent refuses, it contacts the next best one. If normative advice is received, then the agent may move towards the norm gradually. In this mechanism, we have two levels of networks. The lower level is the interaction network or the social link layer. The leadership network evolves on top of this network. At the end of the interactions between agents (i.e. the end of each iteration), norm emergence is studied².

Algorithm 4.1: Pseudocode of an agent to find a normative advisor

```python
normativeAdvisor ← null;
1. Interact with all the agents in the neighbourhood;
2. Compute performance scores;
3. Construct a list (NList) of agents in the neighbourhood ranked by descending order of performance;
4. foreach agent X ∈ NList do
5. normativeAdvice = getNormativeAdviceFromAgent(X);
6. if normativeAdvice != null then
7. normativeAdvisor ← X;
8. Decide whether to make a normative change recommended by the normativeAdvisor;
9. break;
10. end
11. end
```

Figure 4.1 depicts the two network layers that are used in our mechanism. The circles represent agents. The solid lines represent the social link network. The dotted line with an arrow represents the role model relationships that emerges at the end of the interactions. A1 is the leader for A2, A3, A4 and A5. Arrows from these four agents point to A1. This new kind of network that emerges on top of the social link network is called a leadership network. Note that A3 is the leader of A7. A8 is the leader of A1 and A10. A9 is the leader of A8. So,

²Refer to the definition of norm emergence in Section 3.1.5.
the leader at the top of the hierarchy is A8. It should be noted that many agents are leaders and followers at the same time. For example, A1 follows A8 and in turn is followed by four other agents.

Figure 4.1: Two layers of networks used in role model agent mechanism

The next section provides a background on the network topologies on which the role model mechanism for norm emergence has been evaluated.

4.3 Social network topologies

In this section we describe three network topologies that we have considered.

4.3.1 Fully connected network:

In the fully connected network topology, each agent in the society is connected to all the agents in a given society as shown in Figure 4.2 (a). Many multi-agent researchers have done experiments with this topology. Most of their experiments involve interactions with all the agents in the society (Boman, 1999; Verhagen, 2000).
4.3.2 Random network:

Erdős and Renyi have studied the properties of random graphs and have demonstrated a mechanism for generating random networks (Erdős and Renyi, 1959). An undirected graph \( G(n,p) \) has \( n \) vertices in which the edges are connected to each other with a probability \( p \). The graph shown in Figure 4.2 (b) is a random graph with 20 vertices and the probability that an edge is present between two vertices is 0.2. It should be noted that the random network becomes fully connected network when \( p=1 \).

4.3.3 Scale-free network:

Nodes in a scale-free network are not connected to each other randomly. Scale-free networks have a few well connected nodes called hubs and a large number of nodes connected only to a few nodes. This kind of network is called scale-free because the ratio of well connected nodes to the number of nodes in the rest of the network remains constant as the network changes in size. Figure 4.3 is an example of an Barabasi-Albert scale-free network where the size of the network is 50.

Barabasi and Albert (Barabasi and Albert, 1999) have demonstrated a mechanism for generating a scale-free topology based on their observations of large real-world networks such as the Internet, social networks and protein-protein interaction networks (Mitchell, 2006). They have proposed a mechanism for generating scale-free networks based on the
preferential attachment of nodes. At a given time step, the probability (p) of creating an edge between an existing vertex (v) and the newly added vertex is given by the following formula:

$$p = \frac{\text{degree}(v)}{|E| + |V|}$$

where $|E|$ and $|V|$ respectively are the number of edges and vertices currently in the network (counting neither the new vertex nor the other edges that are being attached to it).

One may observe that the network shown in Figure 4.3 has a few well connected nodes, which are called hubs, e.g. vertices V7, and V1. A large number of nodes are connected to very few nodes. Scale-free networks exhibit a power law behaviour (Barabasi and Albert, 1999) where the probability of the existence of a node with k links ($P(k)$) is directly proportional to $k^{-\alpha}$ for some $\alpha$.

### 4.3.4 Some characteristics of networks:

Researchers have studied several characteristics of networks such as diameter (D), average path length (APL), degree distribution (k) and clustering coefficient (C). For our experiments we have used three of these characteristics whose definitions are given below.
• Degree distribution (k): The degree of a node in an undirected graph is the number of incoming and outgoing links connected to particular node.

• Average Path Length (APL): The average path length between two nodes is the average length of all possible paths between two nodes.

• Diameter (D): The diameter of a graph is the longest path between any two nodes.

In this work we used the Java Universal Network/Graph Framework (JUNG) to create the three types of network topologies (O’Madadhain et al., 2005). We note that Kawachi process (Kawachi et al., 2004) can also be used to construct different types of network topologies by changing a single parameter.

4.4 Interaction context and the experimental set-up

In this section we describe the context of interaction between agents in the society and the details about the experimental set-up.

4.4.1 Interaction context

We have experimented with agents that play the Ultimatum game (Slembeck, 1999). The agents are aware of the rules of the game. This game has been chosen because it is claimed to be sociologists’ counter argument\(^3\) to the economists’ view on rationality (Elster, 1989b). When agents interact with each other, their personal view of a norm (\(p\text{-norm}\))\(^4\) may change. Their individual p-norms may tend to change in such a way that it appears that there is a commonly agreed group norm (\(g\text{-norm}\)) in the society.

\(^3\) Sociologists consider that the norms are always used for the overall benefit of the society. Economists on the other hand state that the norms exist because they cater for the self-interest of every member of the society and each member is thought to be rational (Gintis, 2003). When the ultimatum game was played in different societies, researchers have observed that the norm of fairness evolved (cf. Nowak et al. (2000)). As the players in this game choose fairness over self-interest, sociologists’ argue that this game is the counter argument to economists’ view on rationality.

\(^4\)Note that \(p\text{-norm}\) in this thesis is the short form of personal norm. This is different from Tuomela’s prudential norm (also shortened to p-norm).
Note that there is a distinction between a personal norm (\textit{p-norm}) and a group norm (\textit{g-norm}). A \textit{p-norm} is the personal value of an agent. For example an agent may consider that littering is an action that should be prohibited in a society. An agent may recommend this \textit{p-norm} to others. However, this personal value may not be shared by the other agents in a society. A \textit{g-norm} is a norm that an agent believes is the group norm of the society (i.e. that every other agent in the society is expected to follow the norm). While the \textit{p-norm} is private to an agent, an agent assumes that \textit{g-norm} is known to all the agents in the society. However, these two types of norms are internally stored in an agent (i.e. they are not publicly visible).

\subsection{4.4.1.1 The Ultimatum game}

The Ultimatum game (Slembeck, 1999) is an experimental economics game in which two parties interact anonymously with each other. The game is played for a fixed sum of money (say $x$ dollars). The first player proposes how to share the money with the second player. Say, the first player proposes $y$ dollars to the second player. If the second player rejects this division, neither gets anything. If the second accepts, the first gets $x-y$ dollars and the second gets $y$ dollars.

\subsection{4.4.2 Description of the multi-agent environment}

The experimental set-up of an agent society is made up of a fixed number of agents. They are connected to each other using one of the three social network topologies (fully connected, random or scale-free)\textsuperscript{5}.

\subsection{4.4.2.1 Norms in the society}

Each agent in a society has a personal norm (\textit{p-norm}). A \textit{p-norm} represents an agent’s opinion on what the norm of the society should be. In the context of the Ultimatum game, the social network topologies were chosen because they reflect different social structures. Fully-connected network topology was chosen for investigation because it represents the structure of a close-knit society where everyone knows every other person. Random network was chosen because it is a standard topology against which most other network models are compared. Scale-free network topology was chosen because this topology is found in large real-world networks such as the Internet. There are other topologies such as the small-world networks (Watts, 1999) which can be considered in future evaluations.

\textsuperscript{5}Fully-connected network topology was chosen for investigation because it represents the structure of a close-knit society where everyone knows every other person. Random network was chosen because it is a standard topology against which most other network models are compared. Scale-free network topology was chosen because this topology is found in large real-world networks such as the Internet. There are other topologies such as the small-world networks (Watts, 1999) which can be considered in future evaluations.
the norm encodes the maximum and minimum proposal and acceptance values of an agent. A sample $p$-norm for an agent is given below where $min$ and $max$ are the minimum and maximum values when the game is played for a sum of 100 dollars.

- Proposal norm (min=1, max=30)
- Acceptance norm (min=1, max=100)

The representations given above indicate that the proposal norm of an agent ranges from 1 to 30 and the acceptance norm of the agent ranges from 1 to 100.

The proposal norm initialized using a uniform probability distribution within a range of 1 to 100, is internal to the agent. It is not known to any other agent. The agents in a society are initialized with an acceptance norm that indicates that any agent which proposes within the range specified by the norm will be accepted. The agents are only aware of their own acceptance norms and are not aware of the acceptance norms of the other agents. In order to observe how proposal norms emerge, we assign a fixed value for acceptance norm to all the agents in the society. The acceptance norm of a society is given below $^6$.

- Acceptance norm (min=45, max = 55)

Each pair of agents play a certain number of games. In each game, the roles (proposer, acceptor) are randomly assigned. The proposing agent draws a value from the range between the maximum and minimum values associated with its proposal norm. If the value proposed by the proposing agent falls within the acceptance range of the acceptor agent, then the proposal is accepted. In this case, the success count of the proposing agent increases by one. The percentage of success of the proposer agent is known to other agents that are connected to it.

4.4.2.2 Autonomy of agents

Autonomy is an important attribute of software agents. According to Verhagen (2001), autonomy in the context of norms means the freedom of choosing not to comply with the

---

$^6$This is to model a fair society with a deviation of plus or minus five from the equitable division of 100 between two agents.
norms. In our work we extend the notion of autonomy to the recommendation of normative advice. An agent, when asked for normative advice, can decide not to provide advice. An agent can also decide not to take into account the normative advice received (i.e. ignore the normative advice).

In our experimental set-up, when an agent is created, it has an autonomy value between 0 and 1. Depending upon the autonomy value, an agent can either accept or reject a request from another agent. If the autonomy value of an agent is 0.4, it will reject the request from another agent 4 out of 10 times. Once rejected, an agent will contact the next best performing agent amongst its neighbours. Autonomy of an agent is also related to accepting or rejecting the advice provided by the leader agent.

Assume that agent A and B are acquaintances (are connected to each other in a network). If agent A’s successful proposal average is 60% in the game (i.e. A’s proposal is accepted 6 out of 10 times) and agent B’s successful proposal average is 80%, then agent A will send a request to agent B asking for its advice. If agent B accepts this request, B becomes the role model of agent A and sends its $p$-norm to agent A. The agent is autonomous to choose or ignore the advice depending upon its autonomy. When agent A decides to follow the advice provided by agent B, it modifies its $p$-norm by moving closer to the $p$-norm of agent B$^7$.

### 4.4.2.3 Interaction between agents

In our mechanism, an agent plays a fixed number of Ultimatum games with each of its neighbours (agents that are linked to it). In total, highly connected agents play more games than the poorly connected agents. Highly connected agents benefit from playing more games because they retain their competitive advantage of obtaining a wide range of information or norms from the agents that they are connected to, while the poorly connected agents rely on the information from one or two agents that they are connected to. A highly connected agent

---

$^7$In theory, what an agent thinks ought to be a social norm might be different from what its $p$-norm is. For example, an agent may expect that the other agents in the park should not litter, however, it may not want to abstain from littering. In this work, we do not model this dichotomy. We assume that an agent believes that its $p$-norm might be beneficial to the society (i.e. $p$-norm is the ought to be social norm). Hence, it recommends its $p$-norm to other agents.
is more likely to know about the best norm\textsuperscript{8} earlier than the poorly connected agent\textsuperscript{9}.

![Two layers of networks used in role model agent mechanism](image)

**Figure 4.4**: Two layers of networks used in role model agent mechanism

After one iteration, every agent looks for the best performing player in its neighbourhood. After finding the best performing player, the agent sends a request to the player requesting the agent to be its role model or leader. If the requested agent decides to become the role model, it sends its \textit{p-norm} (normative advice) to the requester (follower agent). The follower agent modifies its norm by moving closer to the role model agent’s norm. If an agent does not find a role model agent, it does not change its norm in that iteration.

Figure 4.4 shows an example of the leadership network that emerges on top of the social link network. It also shows how the norms emerge on top of the leadership network. The value that appears inside the circle (in parentheses) is the proposal norm of an agent. An agent will be successful if its proposal norm is between 45 and 55 as the acceptance norms have been fixed to these values. It can be observed that the leaders in the top hierarchy have values that fall in the range of the acceptance norms (i.e. A9 has a value of 50, A8 has a

\textsuperscript{8}The best norm corresponds to the highest proposal value.

\textsuperscript{9}This is akin to powerful people (hubs) knowing important information fast through the connections they possess.
value of 49 and A1 has a value of 48). It should be noted that to start with, the leadership network may not be fully connected. For example, let us assume that A10 starts with a proposal value of 30. In the first iteration this agent will not be successful in its proposal (0% success). In this case, the leadership link between A10 and A8 will not exist. In the same iteration, assuming that agent A8 has a proposal value of 48, A10 will choose A8 as the leader. After a number of iterations A10 will move towards the norm of A8. Once A10 starts being successful (i.e. when A10 reaches a value of 45), A6 will choose A10 as the leader. A6 will then gradually start moving towards the norm of A10.

It should be noted that at end of each iteration a new leadership network emerges. This process is repeated for certain number of iterations. At the end of each iteration, we study the emergence of norms.

### 4.5 Experiments and results

In this section we present the experiments that we undertook to demonstrate that our mechanism leads to norm emergence\(^{10}\) when tested on top of different kinds of network topologies. Unless stated otherwise, each experimental result presented in this thesis is based on a sample run of the system and the result can be reproduced by retaining the same values for the parameters used.

#### 4.5.1 Demonstration of the emergence of leadership networks and norms

Figure 4.5 shows a series of snapshots on the emergence of leadership networks. There are 4 snapshots in total. The top-left snapshot shows the structure of the social network created using Erdős-Rényi (ER) random network with \(p=0.2\). The role model mechanism

---

\(^{10}\)Researchers have different notions of the success of norm emergence. For example, Kittock (Kittock, 1994) considers a norm to have emerged if the convergence on a norm is 90%. In our case we have 100% convergence as our target for norm emergence because we are interested in investigating the conditions under which the norm spreads to the entire society as opposed to a partial spread of the norm. However, we note that a norm is considered to exist when it is more prevalent than any of the competing norms. In theory, the convergence value could be any positive number as long as its observed frequency is greater than that of the competing norms.

92
Figure 4.5: Snapshots of the emergence of leadership and norms in an agent society

is applied on top of this network. At the end of iteration 1, the leadership network starts to appear (top-right). It can be observed that there are two leaders A3 and A6\textsuperscript{11}. A3 has larger number of followers than A6. In iteration 2 (bottom-left), it can be observed that there is an additional leader in the network A7. A7, the follower of A3 has five other followers. In iteration 3 (bottom-right), more followers can be observed. A9, the follower of A7 has one other follower, A1. It should be noted that the proposal values of the followers move towards the proposal values of the leaders. This demonstrates that the leadership network emerges

\textsuperscript{11}The agents referred to are highlighted by dotted green circles. The minimum proposal values of agents are shown in the figure.
on top of the social link network and also the spread of norms in the leadership network.

4.5.2 Norm emergence on top of random and scale-free networks

The role model agent based mechanism for norm propagation was evaluated using Erdős-Renyi (ER) random network and Barabasi-Albert (BA) scale-free network.

![Norm convergence in ER networks](image)

**Figure 4.6:** Norm convergence in ER networks when the average degree of connectivity is varied

At first we studied the effects of changing the average degree of connectivity ($<k>$) on norm emergence, while maintaining a constant population size (N). The average degree of connectivity represents how connected the agents in the society are. A higher value of $<k>$ represents a well connected network. We varied the degree of connectivity ($<k>$ = 5, 10, 20, 100, 200) for the ER and BA networks with N=200. It can be observed from Figure 4.6 that as $<k>$ increased the rate of convergence increased in ER networks. When $<k>$ is 10, 100% norm emergence was observed in the 6th iteration while it only took 3 iterations for convergence when the value of $<k>$ is 200. Note that when $<k>$ equals N, the network is fully connected, hence the convergence is faster.

The comparison of ER and BA networks for the same values of N and $<k>$ is shown in Figure 4.7. It can be observed that there is no significant difference in the rate of convergence
in ER and BA networks. Our experimental results on norm convergence are in agreement with the statistical analysis carried out by Albert and Barabasi on the two kinds of networks (Albert and Barabasi, 2002). They have observed that the diameter (D) and average path lengths (APL) of both the networks are similar for fixed values of N and \(<k>\). The diameters of ER and BA networks, when N and \(<k>\) are fixed are directly proportional to \(\log(N)\). As the diameters of both the networks are the same, the rate of norm convergence are similar.

The parameters D and APL of these networks decrease when the average connectivity of the network increases. When the average connectivity increases, it is easier for an agent to find a leader agent whose performance scores are high. If the average connectivity is low, it would take an agent a few iterations before its leader obtains the norm from a better performing agent. This explains why norm convergence is slower when average connectivity \(<k>\) decreases (shown in Figure 4.6).

Even though the norm emergence properties of both kinds of networks are comparable, it has been argued that the scale-free network is better suited to model norm propagation because in the real world (Albert et al., 2000), people are related to each other through the social groups that they are in, such as work groups and church groups. Information percolates among the members of the group through interactions. Also, people seek advice from a close
group of friends and hence information gets transmitted across social networks. Other researchers have demonstrated that scale-free networks are well suited to explain mechanisms of disease propagation and dissemination of ideas (Mitchell, 2006). Scale-free networks are more robust than random networks when random nodes start to fail (because the probability of the largest hub node failing at random thus fragmenting the network is small) but are susceptible to non-random attacks (i.e. targeted attacks on hubs). This phenomenon has been observed in real world networks (Albert et al., 2000; Albert and Barabasi, 2002).

Bollobás and Riordan (2004) have observed that the diameter and average path lengths of an BA network depends upon the value of $m$, a parameter that can be set to specify the number of nodes to which a new node entering the network should be connected to, using the preferential attachment scheme. When $m=1$, $D$ and $APL$ are directly proportional to $\log(N)$ and for $m>1$, $D$ is directly proportional to $\log(N)/\log(\log(N))$. In this light, Barabasi and Albert have suggested that the scale-free networks should be more efficient in bringing nodes closer to each other, a feature which makes them suitable for propagation of ideas and norms.

![Figure 4.8: Power law behaviour of the leadership network](image_url)
4.5.3 Power law behaviour of the leadership network

We have also observed that the leadership network that emerges on top of the BA network follows a power law behaviour. It is interesting to note that the leadership network that emerges on top of an ER network follows a power law behaviour when the average degree of connectivity is small. For smaller probabilities (p=0.05, 0.1) we have observed that there are fewer leader agents with large number of followers and a large number of leaders with a few followers. Figure 4.8 shows the log-log plot of leaders with k followers identified on the x-axis and the number of leaders with k followers (N(k)) divided by the number of leaders with exactly one follower (N1) identified on the y-axis. The trendline shows the approximate power law behaviour of the leadership network. The average slope of the power law curve was found to be -1.6. Our results are in agreement with that of Anghel et al. (Anghel et al., 2004) who studied the emergence of scale-free leadership structures using minority game (Challet and Zhang, 1998). In their work, an agent sends its game strategy to all the agents in its neighbourhood. There is no explicit notion of leadership as each agent maintains an internal model of who its leader is. In our work, each agent chooses its leader explicitly and the leader sends the norms only to its followers.

4.6 Discussion

Our work is different from other works in this area since we use the concepts of oblique transmission (Boyd and Richerson, 1985) in the mechanism we have proposed. Verhagen’s thesis (Verhagen, 2000) focuses on the spreading and the internalization of norms. This assumes that a top level entity (say, a Normative Advisor) knows what the norm should be and this group norm (g-norm) does not change. The agents can choose to ask for normative advice from the normative advisor. The g-norm is spread to the agents through the normative advice provided using a top-down approach by a centralized authority (i.e. there is only one leader in the society). Our work differs from this work insofar as we employ a bottom-up approach for the emergence of norms. In our work we make use of distributed leadership. Our work allows for a hierarchy of leaders (or distributed leadership) which is similar to what is observed in real world politics. For example, a local leader can have followers. At
the same time, the local leader may be following the national leader. Another contribution of our work is the consideration of different network topologies which cater for modeling different types of network structures that may be present in agent societies.

We have considered a single dimensional attribute for a norm (i.e. proposal value), but in general, agents can be influenced by others based on multi-dimensional attributes. We believe, multi-dimensional attributes can be added to our mechanism (e.g. works on opinion dynamics (Hegselmann and Krause, 2002; Fortunato et al., 2005)). In real-life a human agent can be inspired by two different role models in a particular context and can borrow different attributes of these role models. Additionally, an agent can have different role models for different contexts. These extensions can be investigated in the future.

The role model mechanism does not explicitly consider sanctions (i.e. one may argue that the model addresses convention emergence rather than norm emergence). We note that conventions are also treated under the umbrella of norms both in sociology (e.g. works on coordination norms (Bicchieri, 2006)) and multi-agent systems (Shoham and Tennenholtz, 1992; Sen and Airiau, 2007). We note that the role model mechanism can be extended to include sanctions by allowing the leaders to sanction other agents who do not follow the norm. We believe, the effect of this would be a faster convergence towards norms.

In this chapter we have experimented with a role model based mechanism on top of static network topologies (i.e. the topology of the network does not change). We are interested in experimenting with scenarios where dynamic network topologies are created by agents joining and leaving the network. Additionally, we are interested in investigating how two or more societies can be brought together and in experimenting with scenarios in which different norms may co-exist for the same context (e.g. different proposal norms in the society when the ultimatum game is played). These form the focus of the next chapter.

4.7 Summary

The main contribution of this chapter is the role model based mechanism for norm emergence. We have described a mechanism based on the concept of role models for norm emergence in artificial agent societies. Through experimental results we have demonstrated how norms may emerge in an agent society using the role model mechanism (i.e. oblique norm
transmission mechanism). Our mechanism was tested on top of three network topologies. We have also demonstrated how a hierarchy of leaders emerge on top of a social network topology, which is akin to a real-world human society.
Chapter 5

Norm emergence in dynamically changing networks

5.1 Introduction

Most works on norm emergence on top of network topologies are limited by the fact that researchers have only considered statically created network topologies (Verhagen, 2000; Sen and Airiau, 2007; Villatoro and Sabater-Mir, 2009). To the best of our knowledge very few multi-agent researchers have considered the role of dynamic network topologies for norm emergence. In the real world, people join and leave social groups. In this process, they constantly acquire and lose links (e.g. in circumstances such as switching jobs and migrating from one country to another) and the arrival and the departure rates of the agents may be different. This process leads to the creation of dynamically changing network structures (Albert and Barabasi, 2000; Dorogovtsev and Mendes, 2002).

In the previous chapter, we presented how norms emerge on top of three different types of networks. These networks were static and did not change over time. In order to accurately model what goes on in the real world, the role model mechanism for norm emergence should be tested on top of dynamically changing network topologies.

In order to test the role model mechanism on top of dynamically changing networks, the dynamic networks should be created first. To construct dynamic networks, we have adopted a model representing agents as particles colliding in a social space (Gonzaléz et al., 2006b).
We construct a series of dynamic networks and then test the role model mechanism on top of these networks.

The chapter is organised as follows. In Section 5.2 we present the architecture of the experimental set up that constructs the dynamic networks. In Section 5.3 we describe the experiment we conducted on top of a dynamic network. Section 5.4 presents the extension we have made to the model that constructs the dynamic networks to include how two networks can be brought together. In Section 5.5 we present the experiments we have conducted on top of the two networks before and after merger. The discussions relevant to this chapter are presented in Section 5.6.

5.2 Architecture of the experimental setup for norm emergence

The architecture of our experimental set up for norm emergence consists of two components, the social network topology and the role model mechanism. As shown in Figure 5.1, the networks are constructed using the mobile agent model of González et al. (2006b) (described in Section 5.2.1).

![Figure 5.1: Architecture of the experimental setup for norm emergence](image)

$N_0$ represents the snapshot of a network at time 0. The network is then perturbed by changing the links (adding and deleting links). This results in $N_1$ at time 1. Using this process we create a set of networks. Once the system that creates the networks reaches a stable state called Quasi-Stationary State (QSS) (described in subsection 5.2.1), say at time $t$, we start recording the network topologies. As the second step, we then apply the role model...
mechanism for norm emergence on each of the network structures starting from $N(t+1)$ and study the emergence of norms.

5.2.1 Mobile agent model

González et al. (2006b) have developed a model for constructing dynamically changing networks. In their model they have used the concept of colliding agents (or particles) to construct evolving networks. There are $N$ agents distributed in a two dimensional system (square box) of linear size $L$ which represents an abstract social space. Each agent has the same diameter $D$. Each agent is initialized with a random starting position. Every agent has the same initial velocity.

When two agents collide, they form a link. This link represents a connection between the agents in a social space rather than a connection in the physical space. After each collision both agents move in a random direction. The velocity of each agent is directly proportional to its number of links or contacts ($k$). The more links an agent has, the faster it travels and hence it has an increased ability to attract more links. This phenomenon is observed in real world sexual networks where individuals with a larger number of partners are more likely to get more partners (Lilijeros et al., 2001). The velocity of agent $x$ is given by the formula given below.

$$v_x = v_{x(init)} \cdot k \cdot \omega$$

In the equation shown above, $v_{x(init)}$ is the initial velocity of the agent, $k$ is the total number of links acquired by agent $x$, and $\omega = (e_x \cos \theta + e_y \sin \theta)$ where $e_x$ and $e_y$ are unit vectors and $\theta$ is a random angle that governs the direction of movement. It should be noted that the momentum is not conserved in this approach.

Each agent has a lifetime that is a random number drawn from a uniform distribution in the interval between 0 and a maximum Time-To-Live (TTL). The maximum TTL value is also referred to as relationship duration which represents the maximum time allowed for an agent to create links with the other agents in the society. Once the agent has lived up to a time equal to its maximum TTL, it will be replaced by a new agent with zero links.

Figure 5.2 shows the snapshot of the visualization of the mobile agent model. The simulation panel on the left shows the mobility of the agents and the collisions. The network panel
on the right shows the dynamic construction of network topologies for the agents shown on the left panel. Note that these two panels are displayed concurrently.

Each iteration corresponds to one time step. The simulations can be conducted by varying the parameters such as number of agents (N), maximum Time To Live (TTL) and the size of the square box (L). Gonzalez et al. have observed (González et al., 2006b) that the system reaches a Quasi-Stationary State (QSS) when the number of simulation steps is around twice the value of TTL ($t \approx 2\times$TTL). At the QSS, the average degree of connectivity $<k>$ of the network (total number of links (M) divided by N) is almost constant (very little fluctuations). Networks that are obtained after the system reaches the QSS are considered stable. Hence, any experimentation on network topologies should only consider those networks that are obtained after QSS is reached.

Our implementation is based on that of Gonzalez et al. (González et al., 2006b), but the difference is that they have used continuous time whereas we have used discrete time steps to carry out our simulations.

5.3 Norm emergence in a single agent society

In this experiment we demonstrate how norms emerge in a single agent society that is constructed based on dynamically changing networks. The society consists of 100 agents, with
their social space simulated by a square box of linear size \( L=500 \). The density of the system \( (N/L^2) \) is \( 0.0004 \). Once the dynamically changing network reaches the QSS (at a time step greater than twice the TTL), we took snapshots of the network structure for the next 300 time steps. These 300 snapshots depict the evolution of dynamically changing networks as shown in step 1 of Figure 5.1.

We then apply the role model mechanism that we have designed on top of these 300 networks sequentially. The agents in the network interact with each other in the context of playing the ultimatum game (Slembeck, 1999) for a sum of 100 dollars. The set-up for interactions among agents is the same as what has been discussed in Chapter 4. Each agent has a particular range for acceptance values (say 45 to 55) and proposal values (say 30 to 40). These ranges of values represent the acceptance and proposal norms. By fixing a particular range for the acceptance norm (say 45 to 55) in the society, we observe how the proposal norm emerges in the society. The agents are initialized with the values for proposal norms based on a uniform distribution. An agent chooses another agent in its neighbourhood as its role model based on the success rate of its proposals.

Figure 5.3 shows the norm emergence when \( N=100 \) for three values of maximum TTL.
which are 50, 100 and 150. It can be observed that the proposal norm values gradually increase to attain a steady state. It can also be observed that when maximum TTL values are 50 and 100, the norm emergence is not 100% because the underlying network does not have a giant cluster which encompasses all the nodes so that a norm could propagate. In other words, there isn’t a path between one node to all other nodes. It has been observed (González et al., 2006b) for collision rates approximately equal to 2.04 a giant cluster starts appearing which indicates that there exists a path between any two nodes. When TTL=150, the collision rate exceeds this threshold and hence 100% norm emergence is observed.

5.4 Extending mobile agent model for bringing two societies together

![Figure 5.4: A dynamically constructed network with 50 mobile agents in two groups before and after the collapse of the wall between them](image)

Let us now imagine that two societies with different norms exist. Assume these two societies come into contact with one another. For example, two societies in the real world could be brought together because of reasons such as earthquakes, famines, floods, war and eco-
nomic instability. In this scenario, it is interesting to study how the norms in these societies emerge.

An important aspect to consider for norm emergence is how we could model the interaction between these two societies and the resultant network topologies. This problem can be addressed by extending the mobile agent model. In the mobile agent model, we can represent the societies by two square boxes and the agents inside each box collide to form a network as shown in the top part of Figure 5.4.

Then, at a particular point in time after the system has reached the QSS, the wall that separates both the societies is removed. When the wall is removed, the agents from one society can interact with the agents in the other society. Initially the mobile agents in the edges of both the boxes collide and over time there is a good dispersion of mobile agents. The bottom part of Figure 5.4 shows a snapshot of two societies coming together based on the mobile agent interaction model.

This approach provides a simple and intuitive model to illustrate how two societies can be brought together. Using this model we can record networks that are created through agent interactions.

5.5 Norm emergence when two societies are brought together

Suppose there are two societies of agents, namely G1 and G2. Each society is expected to evolve a particular proposal norm. The two norms are the selfish norm (S norm) and benevolent norm (B norm). These norms indicate that the agents in the first group will propose a low value of money ($35 to $45 out of $100) to the opponents in an ultimatum game, and the agents in the second group will propose more than a fair share ($55 to $65 out of $100) when playing the ultimatum game. In our experimental set up, these two norms can have any range of values of money between 0 and 100.

Figure 5.5 shows the norm emergence patterns in two societies when the maximum TTL of agents is 100. It should be noted that the agents in G1 can have three kinds of norms (which represent three different ranges of values for the proposal norm), the S norm, the B
norm and also any other norm that is different from the S and B norms. The same holds for G2. Because the initial norm distribution values are assigned using a uniform probability distribution, there will be a small portion of B norm adherers even in the society that has emerged with an S norm. Similarly, there will be a small number of S norm adherers in G2. It should be also noted that we are trying to see if an S norm emerges in G1 and a B norm emerges in G2 and then which of these two norms might emerge after the groups are brought together.

It can be observed from Figure 5.5 that, initially, both groups had a high number of other norm observers (around 70%). As the agents interacted with other agents in the same group, an S norm emerges in G1 and a B norm emerges in G2 before the societies are brought together (before iteration 200). The B norm in G1 and S norm in G2, and the other norms in both the groups have disappeared before the groups are brought together.

When the wall is removed, it can be observed that the norm emergence values oscillate closer to 50% and at around 325 iterations the B norm takes over the S norm in the entire
This norm emergence on top of a given network is the result of the role model mechanism through which each agent in the society chooses a role model. If an agent with S norm chooses an agent with B norm as its role model then that agent will gradually move towards B norm. So, the drivers for emergence are a) the underlying dynamically constructed network and b) the role model mechanism that is applied on top of these network structures. We have also observed that, when the same experiment is repeated, a different norm takes over the society which is based on the underlying dynamically created network.

Figure 5.6: Graph showing the convergence results when two societies are brought together (TTL = 50)

We also experimented with lower values of maximum TTL. We decreased the maximum TTL from 100 to 50. It can be observed from Figure 5.6 that the two groups have not had a complete norm emergence before the wall is collapsed. The S norm emergence in G1 is

\[ \text{Note that after the wall is removed we do not wait for the system to reach its QSS, because interesting norm dynamics takes place during this period of instability. Abstaining from the study of norm emergence during this period is equivalent to not allowing agents from different societies to influence each other until QSS is reached, which is not a realistic scenario (e.g. withholding a new migrant to influence or be influenced by the norms of the new country for a considerable amount of time).} \]
around 70% and the B norm emergence of G2 is around 90% before the two groups interact. There are also some agents with other norms. This partial norm emergence is attributed to the lower collision rates as a result of the lower value for maximum TTL. As the life span of the agents are lower, they form a lesser number of links and hence the norm emergence is not 100%. When the agents in both groups are brought together, different types of norms might co-exist. It can be observed that frequencies of both S and B norms are similar and also there are agents with other norms. Again, this behaviour is the result of the lower number of links that are formed between societies due to lower value for maximum TTL.

![Graph showing the norm convergence results when two societies of different sizes are brought together (TTL = 100)](image)

**Figure 5.7:** Graph showing the norm convergence results when two societies of different sizes are brought together (TTL = 100)

We also studied the effect of relative population sizes when two societies are brought together. We modified the experimental setup in such a way that there were 25 agents in one group and 75 in another. The density of agents in group 1 was 0.0004 and the density of agents in group 2 was 0.000048. This implies that agents in group 1 will interact more frequently than the agents in group 2. The agents in the first group were of the selfish type and they were made to interact more frequently which lead to the creation of a reasonably well-knit society (density = 0.0004). The agents in the benevolent group were made to interact less frequently to simulate a not so well-knit society (density = 0.000048). The agents in
both the societies had the same maximum TTL which was set to 100. It can be observed from Figure 5.7 that the two groups have not had a complete norm emergence before the wall is collapsed. It should be noted that the S norm is the dominant norm in group 1 and the B norm is dominant in group 2 before the collapse of the wall. Overall, the B norm seems to be the dominant one in terms of number of adopters. After the collapse of the wall, the B norm is more prevalent than the S norm, but still has not been spread to the entire population. The reason for the B norm to maintain its lead is due to the network topology. As the agents of both societies interact, they initially interact at the edges, and because the neighbourhood of the agents in society 2 is bigger, the agents from the selfish society are not able to invade and spread the norms. Also, in contrast to the previous two experiments, it should be noted that there are quite a few agents with the other norm after the collapse of the wall. This is because there was a large number of undecided agents before the collapse of the wall.

When we increased the maximum TTL to 200, the societies converged to the norm of the largest society (see Figure A.1 in Appendix A). This was mainly because the agents with the undecided norms (other norms) adopted one of the norms of society after the wall collapsed, and as the larger society had more of these agents the societies converged to the norm of the larger society. When the maximum TTL was decreased to 50, the societies did not converge to any norm because most agents had random norms (other norms) which is attributed to lower collision rates, which resulted in a smaller number of links between agents (see Figure A.2 in Appendix A).

5.6 Discussion

In this work we have investigated how the role model mechanism for norm emergence works on top of dynamically evolving networks. Another contribution of our work is the extension of the mobile agent model to create dynamic networks that depict how two societies can be brought together. Our experimental set-up can be used as a tool to study norm emergence by varying several parameters such as the number of agents (N) and density of agents (ρ), time to live (TTL), initial distribution of norms (uniform, normal etc.) and collision rates.

The network topologies generated by bringing two societies together can be used not only to test how norms emerge but also in opinion dynamics (Hegselmann and Krause, 2002;
Fortunato, 2005), disease spreading (Cohen et al., 2003) etc. Our experimental set-up can be further extended to include bringing more than two societies together and studying how norms might emerge in those scenarios. Also, different mechanisms of norm propagation can be experimented with and validated using our approach.

In the real world, we attach more weight to a particular person’s advice than others. Similarly, the weights of the edges (links) can be considered when the agent makes a decision on who to choose as a role model agent. We plan to incorporate this idea in a future work. Additionally, in our current setting all the links are removed when an agent’s TTL is reached (i.e. an agent who reaches the TTL is replaced by a new agent with zero links). This depicts a scenario where agents do not delete links actively during their lifetime. In real-life, the links are lost during the lifetime of an agent (e.g. agents that move away from a society), so our work could be extended to include dynamic removal of links during the lifetime of an agent.

5.7 Summary

The focus of this chapter was on the evaluation of role model mechanism on top of dynamically created network topologies. We have discussed how a mobile agent model based on collisions in a simulated social space can be used to create dynamic network topologies. We have shown how norms emerge on top of these networks using the role model mechanism. We have also discussed how the mobile agent model is extended to bring two societies together. We have also shown the norm dynamics (e.g. co-existence of norms) in the newly formed agent society.

---

2We note this may involve further extensions to mobile agent model (e.g. creating a simulated model of real-world sexual network by allowing agents to choose partners from the links that they have acquired (González et al., 2006a) which are scale-free) which can be used to model and study specific phenomenon such as spreading of diseases (Cohen et al., 2003) and gossips (Lind et al., 2007).
Chapter 6

Architecture for norm identification

6.1 Introduction

One of the research areas in norms that has not received much attention is norm identification. Most works on norms assume either a powerful leader knows about the norm (Boman, 1999; Verhagen, 2000) or a centralized institution proposes and imposes norms (Arcos et al., 2005a; García-Camino et al., 2006a) in a society. Open and scalable agent societies call for mechanisms which can facilitate agents to identify norms. Hence, a key issue addressed in this thesis is norm identification. To that end this chapter provides a higher level overview of an architecture for norm identification. Chapters 7, 8, and 9 make use of the architecture discussed in this chapter to identify prohibition, obligation and conditional norms respectively. The acronyms used in these chapters and their corresponding expansions are given in Table B.1 in Appendix B.

6.2 The need for a norm identification framework

In order to motivate the technical contribution of this chapter, we present the following scenarios from open agent societies.
6.2.1 Scenario 1 - Virtual environments such as Second Life

Divya Chandran, a new Second Life resident wants to explore the rich interactions offered by the new medium. She wants to go to a virtual park and relax by the fountain and listen to chirping birds. She flies to the virtual park and sees people enjoying the sun. She notices some water fountains and some soft-drink fountains from the sponsor of the park. She would like to get a drink, but does not know if there is a norm governing the usage of the fountain. She wonders if she should get a cup from the jazzy sponsor’s booth by paying money or if she needs to acquire the skill of making the cup object. Once she fills the cup with the drink, can she enjoy her drink in all areas of the park or is she restricted to a particular zone? And finally, what should she be doing with the empty cup? What is the norm associated with littering in the park? Can she drop it anywhere for an autonomous robot to collect it once the cup is dropped or should she find a rubbish bin and drop it? Will the norm of this park be applicable to all the parks in Second Life? When she visits the park at a later date will the norm of the park still be the same?

6.2.2 Scenario 2 - e-commerce environments

Rita Ranjan, a casual user of an auction website wants to participate in an auction. She knows the type of auction she is participating in (e.g. English auction, Dutch auction (McAfee and McMillan, 1987)). But she is not aware of the norms associated with the auction. For example, it may not be apparent to her that it is forbidden in a particular auction house to bid for a fourth item of a particular type after winning three consecutive items of the same type.

6.2.3 Scenario 3 - Massively multi-player online games

Tom a veteran player and Tanya a novice player meet in an online game playing platform and choose to play the game of dragon slaying. Let us assume that no formal arrangements are made between the players. Let us also assume that there exists a norm of reciprocity, where a player who has helped another player to escape from a dragon expects the other player to help him/her escape from the dragon by distracting it if need arises. Being a new
player Tanya does not know about the norm. At some point in the game Tanya might be sanctioned by Tom for not following the norm.

The important question the scenarios described above pose is how will the participants (agents) come to know the norms of the society that they currently belong to (or the norms of the context they are in). Knowing the norms is important to the agents, because the agents can protect themselves from negative consequences (e.g. sanctions or a decrease in reputation). In a nutshell, the research question is what capabilities should be built into an agent to recognize a norm? The internal agent architecture described in this chapter aims to address this question.

6.3 The process of norm identification

In this section we provide an overview of the norm identification framework for an agent to infer norms in the agent society in which it is situated. Social learning theory (Bandura, 1977) suggests that new behaviour can be learnt through the observation of punishments and rewards. It has been noted that the observation of agent actions through the process of social monitoring and learning (Conte and Dignum, 2001; Conte and Paolucci, 2001) can be used to identify norms in the society. An agent employing our architecture makes use of social monitoring and learning approaches to identify norms.

Figure 6.1 shows the architectural diagram of the norm identification framework of an agent called the norm engine. An agent’s norm engine is made up of several components. The circles represent information storage components. The rounded boxes represent information processing components, and diamonds represent decision making components, and the lines represent the flow of information between the components.

An agent employing this architecture follows a four-step process.

Step 1: An agent perceives the events in the environment in which it is situated.

Step 2: When an agent perceives an event, it stores the event in its belief base. Belief base is a repository that contains the beliefs of an agent1. An agent has full control of its belief base and thus can add, modify and delete beliefs based on the evidences for these

1The term base is used to represent the repository used by an agent (e.g. an agent’s run-time memory or a persistent storage used by an agent such as a database).
beliefs. The events observed by an observer are of two types: regular events and signalling events. In the context of enjoying a public park, a regular event is an event, such as an agent moving to another location in a park or sitting on a bench. “Special events” are signalling events\(^2\) that agents understand to be either encouraging or discouraging certain behaviour.

\(^2\)Our usage of signalling in this work differs from the views in Biology and Economics. Biologists have observed that animals send signals to indicate that they are a desirable mate (Smith and Harper, 2003). For example, a peacock displays its quality to peahens through its bright plumage and a long ornate tail. Economists have noted that human agents send signals to others that they are credible through some form of signalling (e.g. acquiring a university degree signals that someone has skills for a particular job (Spence, 1973)). In our work,
For example when an agent litters in the park, another agent can discourage the littering action by shouting at the litterer. The signal in this context is the shouting action. We assume that an agent has the ability to recognize signalling events based on its previous experience. Another example of a norm in the context of a restaurant is expectation that the customers should tip the waiter before departing the restaurant. A customer may be sanctioned by the waiter agent. The sanction here could be a yell or shout action\(^3\). Sanctions and rewards are important because they distinguish a norm from a convention (a behavioural regularity).

Step 3: When a special event occurs, the agent stores the special event in the special events base. It should be noted that all events are stored in an agent’s belief base but only special events are stored in the special events base.

Step 4: If the perceived event is a special event, an agent checks if there exists a norm in its personal norm (p-norm) base or the group norm (g-norm) base\(^4\). An agent may possess some p-norms based on its past experience or preference. A p-norm is the personal value of an agent. For example an agent may expect that agents should not litter a park. This personal value may not be shared by the agents in a society. A p-norm may vary across agents, since a society may be made up agents with different backgrounds and experiences. A punisher agent may discourage another agent that violates its p-norm by punishing it. For example, agent A’s p-norm might be that no one should litter in the park. If an agent B violates this p-norm of agent A, then A may punish B. Agent A can be seen as the norm entrepreneur (i.e. an agent who proposes a norm) and we assume that agent A has the power to punish agent B based on the social roles they play in a society. A g-norm is a norm that an agent infers, based on its personal interactions as well as the interactions it observes in the society. An agent infers g-norms using the norm inference component of the framework. The focus of this work is on the g-norm inference.

\(^3\)The sanction can also be any other disapproval gesture.

\(^4\)If the event is not a special event, nothing happens. We have only shown the branching conditions which have some consequences in Figure 6.1.

signalling is special type of event whose occurrence can be interpreted either as a punishment signal (a sanction) or a reward signal. These signals are responses or feedback of the observing agents on the actions performed by the observed agent. While the actions performed by agents themselves are viewed as signals in disciplines such as Biology and Economics, the signals in our work are the feedback of other agents on the actions performed by an agent.
When a special event occurs, an agent may occasionally decide to invoke its norm inference component to identify whether a previously unknown norm may have resulted in the occurrence of the special event. In the context of a park scenario, after observing sanctions between agents over certain amount of time (e.g. agents yelling at one another) an agent invokes the norm inference component to find out what past events may have triggered the occurrence of the special event (i.e. sanctioning event). In other words an agent is interested to find out whether the occurrence of a special event can be explained by the existence of a norm in the society. The invocation of the norm inference component may result in the identification of a $g$-norm, in which case it is added to the $g$-norm base. The two sub-components of the norm inference component are the candidate norm identification component and the norm verification component which are discussed in detail in Section 6.4.2.

The norm inference mechanism can identify two types of norms, prohibition norms and obligation norms. Prohibition norms may be identified by inferring the relevant events that happened in the past (e.g. identifying littering as the action responsible for the occurrence of a sanction). For identifying obligation norms the agent should reason about what events that did not happen in the past are the likely reason for a sanction (e.g. not fulfilling the obligation of tipping in restaurants).

An agent, being an autonomous entity, can also decide not to invoke its norm inference component for every occurrence of a special event but may decide to invoke it only periodically (e.g. in regular intervals of time\(^5\)). When it invokes the norm inference component, it may find a new $g$-norm which it adds to its $g$-norm base. If it does not find a $g$-norm, the agent may change some of its norm inference parameters and repeat the process again in order to find a $g$-norm or may wait to collect more information.

At regular intervals of time, an agent re-evaluates the $g$-norms it currently has, to check whether those norms hold by invoking the norm inference component\(^6\). For example, an

\(^5\)An agent may collect more evidence about sanctions if it waits for certain period of time.  
\(^6\)In human societies norm re-evaluation happens rarely as the norms tend to be largely permanent (e.g. the norms of cooperation and reciprocity). However, some social norms may change (e.g. smoking in a restaurants which was originally permitted is now prohibited) that require re-evaluation on the part of agents. We believe, in virtual environments, the norms can change at a faster rate as the composition of an open agent society changes. Hence, agents need to re-evaluate norms in regular intervals of time. We note that an agent can modify how often it re-evaluates norms using a parameter which can be used to model either frequent or rare
agent re-evaluates whether a norm it has identified earlier still holds after certain amount of time. If the $g$-norm holds it retains the norm. Otherwise, it removes the $g$-norm from the $g$-norm base. The operational details of the norm inference component are explained in Section 6.4. What an agent does with the norms once it has inferred the norms is out of the scope of this thesis. Several other works have contributed to strategies that agents can follow to decide whether to follow a norm (López y López et al., 2002; Meneguzzi and Luck, 2009; Criado et al., 2010). An assumption these works make is that the agents are aware of the norm ahead of time. The present work aims to provide a solution towards bridging the gap of how an agent infers the norm.

When an agent invokes its norm inference component and finds that a $g$-norm does not apply (e.g. if it does not find any evidence of sanctions), it deletes the norm from the $g$-norm base. There could be two reasons why there are no sanctions in the society. The first reason is that there may not be any punishers in the society. When there are no punishers (i.e. norm guardians), agents can delete the norm. Second reason is that all the agents may have internalized the norm (accepted the norm) and are following the norm\(^7\). Hence, there might not be any sanctions. In this case, the norm deletion on the part an agent who has not internalized the norm may have negative consequences for that agent (i.e. the agent can be sanctioned) in which case it can add the norm again through norm inference. Agents may delete norms that apply to a society when they leave a society\(^8\).

\section*{6.4 Overview of the components of the framework}

This section provides an overview of the components of the norm identification framework. The components that will be discussed are a) event storage components and b) the norm inference component. We will describe the role of the components in the context of a park scenario.

\footnote{Note that how an agent internalizes a norm is out of the scope of this thesis. Other researchers have studied how norms are internalized (Verhagen, 2001). The focus of our work is on norm identification.}

\footnote{We have experimented with two types of scenarios where agents remember the norms of the society they leave and where agents remove the norms of the society they leave.}
6.4.1 Event storage components

Let us assume that an agent is situated in a public park. The agents are aware that they are in a park, and interactions happen within the park context. Additionally, each agent knows other related information about the environment, such as the location of the rubbish bins. Let us also assume that a norm against littering does not exist to start with, but a few of the agents have a notion of what an appropriate action should be in a particular circumstance (a \( p\)-norm). In this architecture an agent would first observe the interactions that occur between the agents in the society. The interactions could be of two types. The first type of interaction is the one in which the agent itself is involved and is called a personnel interaction (an action that an agent does in an environment or a message that is exchanged with another agent). The second type of interaction is an interaction between other agents that is observed by an observer agent, referred to as an observed interaction. The agent records these interactions (events) in its belief base. An agent in the society can assume one or more of the three simple roles: a participant (P) that is involved in a personal interaction, an observer (O) that observes what other agents do and a signaler (S) who sanctions or rewards other agents either because the action is against its \( p\)-norm or is against the \( g\)-norm of a group that it previously belonged or currently belongs. We assume that the signalling agent has the social power (Jones and Sergot, 1996; Boella and van der Torre, 2007) to sanction other agents similar to what is observed in human societies (e.g. a jaywalker being reprimanded, a litterer being rebuked by a member of the public). These signals can either be verbal (e.g. yelling) or non-verbal (e.g. shaking head in disapproval).

When the agents move around the park and enjoy the environment, they may become hungry and eat food. Some agents may litter (i.e. drop the rubbish on the ground), and some agents may dispose rubbish in a rubbish bin. The actions that can be performed by an agent are move, eat, and litter. Some agents consider littering to be an activity that should be discouraged (based on their \( p\)-norms or past experience), so they choose to signal other agents through actions such as yelling and shaking their heads in disapproval. We assume that an agent has a filtering mechanism which categorizes actions such as yell and shake-head as signalling actions. Signalling events can either be positive (e.g. rewards) or negative (sanctions). These signalling actions are stored in the special events base. The signalling
agents can be considered as norm proposers. A norm proposer is an agent that comes up
with a norm and recommends the norm to other agents. A proposer may or may not sanction
other agents for not following the norms. An authoritative leader may choose to sanction
agents that do not follow the proposed norm. A norm entrepreneur may choose to sanction
the members of his group for not following the norm. In the off-line design approach there
may be penalties imposed by the system for not following the norm.

We assume that agents observe each other within a certain visibility threshold (e.g. agents
can only see other agents in a certain neighbourhood in a grid environment). An observer
records another agent’s actions until it disappears from its vicinity. When a sanctioning
agent observes the occurrence of an event that is in breach of one of its norms, the agent may
become emotionally charged and perform a sanctioning action, such as shaking its head vig-
orously in disapproval. The observer agent observing this interaction can infer that someone
involved in an interaction may have violated a norm. Even though an observer may know
that a sanctioning event has occurred, it may not know the exact reason for sanctioning (i.e. it
may not know the norm). It will infer norms using the norm identification framework.

In order to understand what an agent stores, let us assume that an agent perceives other
agents’ actions. An event that is perceived consists of an event index, an observed action,
and the agent(s) participating in that event. For example an agent observing another agent
eating will have the representation of \( \text{happens}(1, \text{eat}(A)) \). This implies the observer believes
that at time unit one, agent A performs an action \( \text{eat} \). A sample representation of events
observed by an agent is given in listing 6.1.

\[
\begin{align*}
\text{happens}(1, \text{eat}(A)) \\
\text{happens}(2, \text{litter}(A)) \\
\text{happens}(3, \text{move}(B)) \\
\text{happens}(4, \text{move}(A)) \\
\text{happens}(5, \text{sanction}(B, A))
\end{align*}
\]

(6.1)

An agent records these events in its belief base. Event 5 is a sanctioning event, where
agent B sanctions agent A. The agents have a filtering mechanism, which identifies signalling

\( ^9 \)A norm following agent can also decide to sanction another agent which may increase the rate of rate of
norm spreading which has not been explicitly modelled here.
events and stores it in the special events base. It should be noted that special events, such as *yell* and *disapproval shake*, are categorized by an agent as sanctioning events and they are stored in the special events base under the *sanction* event. We note that recognizing and categorizing an event into a sanction or a reward is a difficult problem. In our architecture we assume such a mechanism exists (e.g. based on an agent’s past experience). Several researchers are working on detecting emotions in artificial agents such as avatars through extracting facial features, messages and emoticons exchanged between participants (Picard, 1998; Neviarouskaya et al., 2009; Koda and Ruttkay, 2009). We assume that a module that infers emotions already exists (e.g. one of the above mentioned approaches may be employed to recognize and categorize events).

### 6.4.2 Norm inference component

The norm inference component of an agent is made up of two sub-components. The first sub-component makes use of our Candidate Norm Inference (CNI) algorithm to generate candidate norms. The CNI algorithm is made up of two sub-algorithms for identifying candidate prohibition and obligation norms respectively. The second sub-component is the norm verification component, which verifies whether a candidate norm can be identified as a norm in the society.

#### 6.4.2.1 Candidate norm identification component

Candidate norms are the norms that an agent considers to be potential candidates to become the norms in a society. The candidate norm identification mechanism identifies two types of norms, prohibition norms and obligation norms\(^{10}\). Identifying candidate norms is a three step process (Figure 6.2). First, based on the observations an agent creates event sequences.

\(^{10}\)We do not model permissive norms in this thesis because they can be easily expressed as extensions to prohibition norms (i.e. permissions are negations of prohibitions). In the world of permissive norms every action is prohibited until explicitly permitted. But this is not the case in most cases in the real world. A reasonable assumption would be that action is permitted unless prohibited. However, we believe permissions are important when considering exceptions to prohibitions. For example you are normally expected to be seated in a lecture theater and remain quiet during the entire lecture. But, you can run out of a lecture theater in the case of a fire.
These are sequences of events that happen between interacting agents. Second, the agent filters event sequences based on a criterion. The criterion is that the agent is interested only in events that precede a sanction or a reward in order to identify a norm. Third, it uses a data mining approach (Han and Kamber, 2006; Ceglar and Roddick, 2006) to identify candidate norms (both prohibition and obligation norms). The identified prohibition and obligation norms are stored in candidate prohibition norm set (CPNS) and candidate obligation norms sets (CONS), respectively.

A higher level overview of identifying obligation and prohibition norms is given in Figure 6.2. The steps involved are represented using rounded rectangles in Column one. Columns two and three show the state information associated with identifying prohibition and obligation norms respectively. The state information includes the resultant products that are created (solid rectangles), and the processes (circles). Each of these three steps produce certain products (or results). The product of a step is used by the subsequent step. A detailed description of these products and processes associated with prohibition and obligation norms identification is provided in chapters 7 and 8 respectively.

Figure 6.2: Overview of the norm inference component
6.4.2.2 Norm verification component

Once the candidate norms have been obtained, the norm verification component verifies whether a candidate norm can be identified as a norm in the society. An agent asks other agents in its vicinity whether a particular norm holds in the society. If the answer is yes, it then adds the norm to its g-norm base. If the answer is no or the other agent does not know whether the norm holds the agent may ask other agents or choose to gather more data.

In our architecture a new agent entering the society does not ask another agent what the norms of the society are. It first uses the norm identification architecture to infer the norms. It then asks for norm verification. The reasons for asking another agent just for verification are two-fold.

1. First, an agent entering a society may not be interested to find out all the norms of a society (an agent might give a long list of norms followed in the society). It may be interested to find the norms in a particular context. An agent has to first infer what the context is (by observing the interactions) and then it can ask another agent in the neighborhood if its inference of a norm is valid (e.g. *Am I obliged to tip in this society?*). In our view this is more effective (in terms of computation and memory required) than asking another agent what the norms are, as there could be a long list of norms that apply, and most of those may not be of interest to an agent. The agent employing the architecture will be able to infer what the potential norms might be. Hence, it can be confident in asking for norm referral, as the actual norm might be one of the candidate norms in its list. The search space for the actual norm has been narrowed by the norm identification process. An agent can also precisely formulate a query for another agent (e.g. *Is it prohibited to litter in this society?*).

2. Second, an agent may not completely trust other agents in an open society. When an agent asks another agent without norm inference, the other agent could potentially lie about a norm. So, an agent may want to make sure that it identifies candidate norms, before it asks for norm verification. This process helps an agent from being misled by the referring agent if it were to ask what the norm is, since it knows that one of the
candidate norms could potentially be a norm. Note that this does not solve the lying problem, since the referrer agent can lie when an agent asks if something is a norm in the society. At the least, the mechanism we use here allows the agent to have a set of candidate norms. We discuss a potential solution to the lying problem in the discussion chapter (Chapter 10).

Detailed descriptions of these two components (candidate norm identification and norm verification) are presented in the context of identifying prohibition and obligation norms in chapters 7 and 8, respectively. Chapter 9 discusses how the norm identification mechanism is used to identify conditional norms. In order to provide a holistic view of our work in the area of norm identification, we defer the discussion on the contributions, limitations, and the future research directions to Chapter 10.

6.5 Summary

This chapter described the motivation for the research on norm identification using three different scenarios. The question how an agent comes to find out what the norms of society are has not received much attention in the field of normative multi-agent systems. To that end, we have proposed a norm identification architecture from the perspective of a single agent (i.e. an internal agent architecture for norm identification). Most researchers agree that there will be some form of sanction or reward once a norm is established (e.g. Elster (1989b); Coleman (1990)). Hence, the notion of a signal (positive or negative action) has been established as a top level entity in our work. We have assumed that even when a norm is being created, the notion of a sanction is important for norm identification. This chapter provided an overview of the framework for norm identification.
Chapter 7

Identifying prohibition norms

7.1 Introduction

In this chapter we demonstrate how an agent can use the norm identification architecture to infer prohibition norms. The Prohibition Norm Identification (PNI) algorithm presented here enables an agent to identify prohibition norms. Using simulations we demonstrate how an agent makes use of the PNI algorithm to identify prohibition norms.

This chapter is organized as follows. Section 7.2 provides an overview of the attributes used by the norm identification mechanism. The two main sub-components in the norm inference component are described in Sections 7.3 and 7.4. They are a) candidate norm identification component and b) the norm verification component. An agent uses the candidate norm identification component to identify candidate norms. It then uses the norm verification component to verify whether a candidate norm can be identified as a norm in the society. In sections 7.5 and 7.6 experimental results on norm identification are presented. Section 7.7 provides a summary of the contributions of this chapter.

7.2 Definitions of attributes of the norm identification framework

The attributes (or the parameters) of the norm identification framework are defined below.

**History Length (HL):** An agent keeps a history of the observed interactions for a certain
window of time, which is represented by the History length (HL) parameter. For example, if HL is set to 20, an agent will keep the events it observes in the last 20 time steps in its memory.

**Event Sequences (ES):** An event sequence is the record of actions that an agent observes in the history. For example the event sequence observed by an agent where HL=5 is given in Table 7.1. In this case, the observer agent observes two other agents, A and B.

**Special Events Set (SES):** An agent has a set of events it identifies to be special. These events are the signalling events. For example, the special event set can contain events such as *yell*, or *shake head in disapproval* (SES = \{ *yell*, *disapproval head-shake* \}). We assume that an agent has the capability to categorize events into two types, sanctions and rewards. For example the actions mentioned above can be identified as sanctioning actions.

**Unique Events Set (UES):** This set contains the number of distinct events that occur within a period of time. For example, a unique events set for the in a park may contain the following events, UES = \{ *eat*, *litter*, *move*, *sanction* \}. Note that the event occurrences are modelled as simple propositions.

**Occurrence Probability (OP):** The occurrence probability of an event E is given by the following formula.

\[
OP(E) = \frac{\text{Number of occurrences of } E}{\text{Total number of events in ES}}
\]

**Window size (WS):** When an agent wants to infer norms, it looks into its history, a certain number of recent events that precede a sanction. For example, if the WS is set to 3, an agent constructs an Special Event Episode (SEE) with three events that precede a special event. Construction of event episodes is described in Section 7.3). It should be noted that an SEE is a subset of ES.

**Norm Identification Threshold (NIT):** When coming up with candidate norms, an agent may not be interested in events that have a lower probability of being a norm. For example, if an agent sets NIT to be 50 (in a scale from 0 to 100), it indicates it is interested to find all sub-episodes of an event episode that have 50% chance of being a candidate norm (i.e. being the reason for generating a sanction).

**Norm Inference Frequency (NIF):** An agent may choose to invoke a norm inference component every time it observes a special event, or may invoke the component periodically.
An agent has a parameter called norm inference frequency (NIF) that specifies what the
time interval between two invocations of the norm inference component is. An agent, being
an autonomous entity, can change this parameter dynamically. If it sees that the norm in
a society is not changing, then it can increase the waiting period for the invocation of the
norm inference component. Alternatively, it can reduce the time interval if it sees the norm
is changing.

### 7.3 Candidate norm identification component

There are two main steps involved in the norm inference process (see Algorithm 7.1). First,
based on a filtering mechanism, special event episodes are extracted from the event sequences
that are recorded by the agent. Second, based on the special event episodes that are extracted,
the Candidate Prohibition Norm Set (CPNS) is generated using a modified version of the
WINEPI algorithm (Mannila et al., 1997).

**Algorithm 7.1**: Overview of the norm inference process

```
1 foreach invocation of the norm inference component do
2     Create special event episodes from event sequences; /* see Algorithm 7.2 */
3     Create candidate prohibition norms; /* see Algorithm 7.3 */
4 end
```

### 7.3.1 Creating special event episodes

An agent records other agents’ actions in its belief base. We call these events that were
recorded in the belief base “event sequences” (ES). An agent has a certain history length
(HL). Let us assume that there are three agents A, B and C. Agent C observes agents A and
B interacting with each other. A sample representation of events observed by agent C is
given in Table 7.1. We assume that agents can perform only one action at any point in time.
It can be observed that agent A eats in time unit 1, litters in time unit 2, and then moves for
the next three time units, while agent B eats in time unit 1, moves for the next two time units, sanctions agent A in time unit 4 and then moves.

Table 7.1: Events observed by an agent

<table>
<thead>
<tr>
<th>Agent A</th>
<th>Agent B</th>
</tr>
</thead>
<tbody>
<tr>
<td>happens (1, eat(A))</td>
<td>happens (1, eat(B))</td>
</tr>
<tr>
<td>happens (2, litter(A))</td>
<td>happens (2, move(B))</td>
</tr>
<tr>
<td>happens (3, move(A))</td>
<td>happens (3, move(B))</td>
</tr>
<tr>
<td>happens (4, move(A))</td>
<td>happens (4, sanction(B, A))</td>
</tr>
<tr>
<td>happens (5, move(A))</td>
<td>happens (5, move(B))</td>
</tr>
</tbody>
</table>

When an agent observes a special event (e.g. sanction), it extracts the sequence of actions from the recorded history (event sequences) that were exchanged between the sanctioning agent and the sanctioned agent. In the example shown in Table 7.1, the observer infers that something that agent A did may have caused the sanction. It could also be something that agent A failed to do might have caused a sanction. In this chapter we concentrate on the former possibility. Agent C then extracts the following sequence of events that took place between A and B.

\[
\{ A, B \} \rightarrow \text{happens}(1, \text{eat}(A)) \rightarrow \text{happens}(2, \text{litter}(A)) \rightarrow \text{happens}(3, \text{move}(A)) \rightarrow \text{happens}(4, \text{sanction}(B, A))
\]

We call the retrieved event sequence that precedes a sanction as the *special event episode* (SEE). To simplify the notation, only the first letter of each event will be mentioned from here on (e.g. e for eat). Thus the event episode for interactions between agents A and B shown above will be represented as

\[
\left( \{ A, B \} \rightarrow e - l - m - s \right)
\]

There might be a few sanctioning events at any given point of time that an agent observes. A sample special event episode list (SEEL) that contains events that are observed by an agent preceding a sanction where WS=3 is given in Figure 7.1.

The pseudocode for creating an special event episode list (SEEL) is given in Algorithm 7.2. For every special event in the event sequence (ES), an agent creates special event episode
(SEE) with $n$ events that precede the special event where, $n=WS$. Special event episode (SEE) is then added to the special event episode list (SEEL). Even though in the example shown in Figure 7.1, we have assumed that an agent considers three events ($n=3$) that precede a signal (a sanction), the value of $n$ can change according to the computational capabilities of the agent.

*Algorithm 7.2:* Pseudocode to create Event Episode List

| Input: | Event Sequence (ES), Window Size (WS) |
| Output: | Special Event Episode List (SEEL) |

1. **foreach** special event in ES **do**
   1. Create special event episode (SEE) from the last $n$ events that precede the special event, where $n=WS$;
   2. Store special event episode (SEE) in special event episode list (SEEL);
2. **end**

### 7.3.2 Generating candidate prohibition norms

The pseudocode for generating the candidate prohibition norms is given in Algorithm 7.3. Algorithm 7.3 is a modified version of the WINEPI algorithm (Mannila et al., 1997), an association rule mining algorithm. Association rule mining (Ceglar and Roddick, 2006) is a well known field of data mining where relationships between items in a database are discovered. For example, interesting rules such as 80% of people who bought diapers also bought beers can be identified from a database. There are several well known algorithms that can be used to mine interesting rules such as Apriori (Agrawal and Srikant, 1994) and WINEPI (Mannila et al., 1997). The WINEPI algorithm analyses event sequences and identifies frequently occurring episodes in a particular window of time.

As norms in this work are inferred from sequences of events that precede a signalling
event, we have used a modified version of the WINEPI algorithm to identify candidate norms. For example an observer may observe that a littering action always happens (occurrence probability = 1) before the occurrence of a sanctioning event. In this case, the WINEPI algorithm can be used to identify the relationship between the littering action and the sanctioning action. We have modified the WINEPI algorithm such that it generates sub-episodes using “permutation with repetition”. The pseudo code of the modification is given in algorithm 7.4.

Algorithm 7.3 works iteratively, with the number of iterations equal to Window Size (WS). The sub-episodes for the first iteration are of length one. The sub episode list (SEL) for iteration one contains all the events in the UES. For example, for the events in Figure 7.1, the SEL at the first iteration will contain events $e$, $l$ and $m$. For each of the sub-episodes in the SEL, the occurrence probabilities are calculated. If the occurrence probability of a sub-episode in the special event episode list is greater than or equal to the norm inference threshold (NIT), the event is added to the Candidate Prohibition Norms Set (CPNS) (lines 7 to 12). For example, if the occurrence probabilities of events $e$ and $l$ are greater than or equal to NIT, then these will be added to CPNS. Each candidate norm is also added to a temporary list which is used for creating the SEL for the next iteration. The SEL for the next iteration ($SEL_{next}$) is created using Algorithm 7.4 and is assigned to $SEL_{current}$. 
**Algorithm 7.3**: Pseudocode to create Candidate Prohibition Norms Set (CPNS)

**Input**: Special Event Episode List (SEEL), Unique Event Set (UES), Window Size(WS), Norm Inference Threshold(NIT)

**Output**: Candidate Prohibition Norms Set (CPNS)

begin

1. CPNS ←− ∅;
2. iterNum ←− 1;
3. Sub-Episode List (SEL_{current}) ←− UES;
4. while iterNum ≤ WS do
5.     SEL_{temp} ←− ∅;
6.     foreach Sub-Episode(SE) in SEL that appears in SEEL do
7.         if iterNum = 1 then
8.             if OP(SE) ≥ NIT then
9.                 CPNS ←− SE, SEL_{temp} ←− SE;
10.            end
11.        end
12.    else
13.        if each event in SE ∈ CPNS and OP(SE) ≥ NIT then
14.            CPNS ←− SE, SEL_{temp} ←− SE;
15.        end
16.    end
17.
18.    Construct SEL_{next} using SEL_{temp};  /* Algorithm 7.4 */
19.    if iterNum < WS then
20.        iterNum ←− iterNum + 1;
21.        SEL_{current} ←− SEL_{next};
22.    else
23.        return CPNS;
24.    end
25. end

end
Each sub-episode in the second iteration will have two events. In the second iteration a sub-episode in SEL will be added to CPNS if two conditions are satisfied (lines 13 to 17). To be added to CPNS,

1. Each event in the sub-episode should already exist in the CPNS.

2. The occurrence probability of the sub-episode in the special event episode list (SEEL) should be greater than or equal to NIT.

In a similar fashion, the algorithm computes all candidate norms. The maximum length of a candidate norm and the number of iterations of algorithm 7.3 are also equal to WS.

<table>
<thead>
<tr>
<th>Algorithm 7.4: Pseudocode to create Sub-Episode List</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Input:</strong> Candidate norms list (tempList)</td>
</tr>
<tr>
<td><strong>Output:</strong> Sub Episode List (SEL)</td>
</tr>
<tr>
<td><strong>foreach</strong> candidate norm in tempList <strong>do</strong></td>
</tr>
<tr>
<td>1 Generate Sub-Episodes (SE) of length n using other candidate norms (allowing repetition of events);</td>
</tr>
<tr>
<td>2 Add each generated SE to SEL;</td>
</tr>
<tr>
<td><strong>end</strong></td>
</tr>
</tbody>
</table>

Let us consider the following scenario that demonstrates how the norm identification mechanism works. Assume that an agent is interested in three events in an event sequence that precede a sanction (i.e. event episodes of length three). Let us assume that NIT is set to 50%, and the unique event set is \{e, l, m\}. As an example let us consider the special event episode list (SEEL) given in Figure 7.1. In the first iteration, SEL_{current} contains sub-episodes of length one which are e, l and m. For each sub-episode an agent calculates the occurrence probability. The occurrence probabilities for e and l are 100% and the occurrence probability for m is 40%. Since events e and l have an occurrence probability greater than NIT, the agent adds these two events to its candidate norm set. For the second iteration, the agent must calculate sub-episodes of length two that have NIT greater than 50%. It uses Algorithm 7.4 to calculate the next sub-episode list (SEL_{next}). For this purpose it uses the
candidate norms that were found in the previous iteration ($SEL_{temp}$). $SEL_{temp}$ in this case contains \{e, l\}.

Algorithm 7.4 creates sub-episodes for the subsequent iteration based on the candidate norms from the previous iteration. In the running example, in the second iteration the algorithm creates sub-episodes of length two, based on sub-episodes of length one. Allowing repetition of events, the algorithm creates the following sub-episodes \{ee, el, le, ll\} and adds them to the $SEL_{next}$\(^1\). Then, the probabilities of these 4 sub-episodes are calculated. Occurrence probabilities of \{ee, el, le, ll\} are \{10\%, 100\%, 50\%, 0\%\}. As NIT is set to 50\%, el and le are added to the candidate norms set. These two sub-episodes will be considered for the creation of SEL for the third iteration using Algorithm 7.4. For the third iteration the contents of the SEL are \{ele, lel\}.

In iteration three, the occurrence probabilities of \{lel, ele\} are \{20\%, 30\%\}. As the occurrence probabilities of the sub-episodes of length three are below NIT these events will not be added to the candidate norms set. As the number of iterations is equal to WS, the algorithm returns the candidate norm set to the agent. In the end, the candidate prohibition norms set will have the following entries whose occurrence probabilities are greater than or equal to NIT: \{e, l, el, le\}. If the agent sets the NIT to 100\% then the CPNS will contain e, l and el. Table 7.2 shows the norms identified at the end of different iterations when the Norm Identification Threshold (NIT) is varied based on the sample special event episodes given in Figure 7.1. It can be noted that the number of candidate norms decrease when the NIT increases.

<table>
<thead>
<tr>
<th>Iteration</th>
<th>NIT=25%</th>
<th>NIT=50%</th>
<th>NIT=100%</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>e, l, m</td>
<td>e, l</td>
<td>e, l</td>
</tr>
<tr>
<td>2</td>
<td>le, el</td>
<td>le, el</td>
<td>el</td>
</tr>
<tr>
<td>3</td>
<td>ele</td>
<td>-</td>
<td>-</td>
</tr>
</tbody>
</table>

It should be noted that algorithm 7.3 is a modified version of the WINEPI algorithm\(^1\). Permutations with repetitions are considered, because an agent does not know whether littering once (l) is a reason for sanction or littering twice (ll) is the reason for the sanction. It could be that littering once may be allowed but an agent littering twice may be punished.

---

\(^1\)Permutations with repetitions are considered, because an agent does not know whether littering once (l) is a reason for sanction or littering twice (ll) is the reason for the sanction. It could be that littering once may be allowed but an agent littering twice may be punished.
(Mannila et al., 1997), an algorithm for mining association rules. Some well known algorithms in the data mining field can be used for mining frequently occurring episodes (i.e. mining association rules) (Agrawal and Srikant, 1994; Mannila et al., 1997). A limitation of the well-known Apriori (Agrawal and Srikant, 1994) algorithm is that it considers combinations of events but not permutations (e.g. it does not distinguish between event sequences \(el\) and \(le\)). WINEPI (Mannila et al., 1997) addresses this issue, but it lacks support for identifying sequences that are resultants of permutations with repetition (e.g. from sub-episodes of length one, e.g. \(e\) and \(l\), the algorithm can generate sub-episodes of length two which are \(el\) and \(le\), but not \(ee\) and \(ll\)).

The modification we have made to the WINEPI algorithm is given in Algorithm 7.4 which can identify candidate norms that are obtained by considering “permutations with repetition” when constructing sub-episodes. We note that algorithm 7.3 can be replaced with any other association rule mining algorithm\(^2\). Hence, it forms the replaceable component of the framework.

Having compiled a set containing candidate norms, the agent passes this information to the norm verification component.

### 7.4 Norm verification

In order to find whether a candidate norm is a norm of the society, the agent asks another agent in its proximity. This happens periodically (e.g. once in every 10 iterations).

When two agents A and B interact, A chooses its first candidate norm (say \(el\)) and asks B if it knows whether \(el\) is a norm of the society. If the response is affirmative, A stores this norm in its set of identified norms. If not, A moves on to the next norm in its candidate norm set\(^3\).

In the case of the running example, the sub-episode \(e\) has the highest probability for

---

\(^2\)We note there is currently a momentum in developing fast and efficient association rule mining algorithms (e.g. Chang (2011)). These algorithms can be used for mining norms in the future.

\(^3\)Other alternative mechanisms are also possible. For example, an agent can verify whether its candidate norms hold by undertaking actions that it observes to be sanctioned (e.g. by dropping a litter). Based on the outcome of tests the agent carries out it can infer what the norms could be. This is a meta-level norm testing mechanism of an agent.
selection, and it is chosen to be communicated to the other agent. It asks another agent (e.g. an agent who is the closest) whether it thinks that the given candidate norm is a norm of the society. If it responds positively, the agent infers \( \text{prohibit}(e) \) to be a norm. If the response is negative, the next candidate norm set is chosen for verification. The agent then asks whether \( l \) is the reason for sanction. If yes, littering is considered to be prohibited. Otherwise, the agent moves on to the next candidate norm. This process continues until a norm is found or no norm is found, in which case the process is re-iterated once a new signal indicating a sanction is generated. When one of the candidate norms has been identified as a norm of the society, the agent still iterates through the candidate norm set to find any co-existing norms. For example, assume that two norms exist in a park, a norm against littering and a norm against eating. Having identified a norm against littering, an agent will iterate through its candidate norm set to find the norm against eating.

Note that an agent will have two sets of norm repositories in its memory: candidate norms and identified norms. Figure 7.2 shows these two sets of norms. Once an agent identifies the norms of the system and finds that the norms identified have been stable for a certain period of time, it can forgo using the norm inference component for a certain amount of time (based on the norm inference frequency (NIF)). It invokes the norm inference component periodically to check if the norms of the society have changed, in which case it replaces the norms in the identified set with the new ones or deletes the norms which are no more applicable. For example, if an agent identifies that the norm against eating does not hold in a society (i.e. there are no sanctions in the society for eating), it removes this norm from its identified norm set\(^4\).

\(^4\)It could be that there are no sanctions in the society because all the agents have internalized a norm (i.e. they abide by the norm). However, the process of norm internalization has not been considered in this work. Norm internalization can depend upon the personality of agents (e.g. social agents, rebellious agents). We note this can be included in our architecture (discussed in Chapter 10). Additionally, norm internalization is a separate issue from the issue of norm identification studied here.
7.5 Experiments on norm identification

In this section we demonstrate how the agents that make use of the proposed architecture are able to infer the norms of the society.

We have implemented a Java-based simulation environment to demonstrate how norms can be identified by agents in a multi-agent society. A toroidal grid represents a social space where agents can move\(^5\). The snapshot given in Figure 7.3 shows different types of agents in a grid environment populated with four different societies. An agent enjoys the park by moving from one location to another. An agent can move in one of the four directions (up, down, left and right). The agents are represented as circles. There are three types of agents in the system: the litterers (L) in light blue, the non-litterers (NL) in light green and the non-littering punishers (NLP) in red. An agent’s visibility is limited to a particular society (i.e. an agent can observe the actions of all the other agents in its society). The letters that appear above an agent specify the agent number and the action it is currently performing. When an agent infers a norm, a solid square appears inside the circle with the same colour as that of the signalling agent (red in this case). The signalling agent is a norm proposer which punishes other agents probabilistically based on its \(p\)-norm. For experiments reported in Sections 7.5.1 to 7.5.5, all the agents make use of the norm inference mechanism.

\(^5\)A toroidal grid is constructed from a rectangular grid (shown in Figure 7.3) by gluing both pairs of opposite edges together. This forms a three dimensional space (a donut shape) where agents move in circles (Weisstein, 2010b).
7.5.1 Experiment 1 - Norm identification and verification

The objective of this experiment is to demonstrate that agents that use the norm inference architecture can generate candidate norms and also identify norms through the verification process. Agents in a society can verify that a certain norm holds in a society by asking other agents in the society. There were 100 agents in the agent society (50 NL, 46 LL and four NLP agents). The NLP agents punished agents that littered.

7.5.1.1 Norm identification

In order to demonstrate that the norm identification component works, we conducted experiments by varying the NIT and keeping all the other parameters constant (HL=20, NIF=5, WS=3). For a particular agent, when NIT was set to 25%, the agent inferred seven candidate...
norms \{e, el, l, le, lel, ee, eel\} whose occurrence probabilities were \{1, 1, 1, 0.75, 0.75, 0.25, 0.25\}. When NIT was set to 50\%, the agent inferred five candidate norms \{e, el, l, le, lel\}. Note that the candidate norms that are identified when NIT was set to 50\% are a sub-set of the candidate norms that were identified when NIT was set to 25\%. When NIT was set to 100\%, the agent inferred three candidate norms \{e, el, l\}.

### 7.5.1.2 Norm verification

In our experimental set-up an agent can ask one other agent in its vicinity (randomly chosen) about a candidate norm. If that agents answers positively, then the agent will promote the norm to the identified norm set. When seeking norm verifications, an agent can use one of the following approaches. It can either ask a) a sanctioning agent or b) any agent that possesses a norm (e.g. the agent may have obtained the norm from a sanctioning agent). Figure 7.4 shows two lines which correspond to these approaches. It can be observed that all the agents in the society identify a norm by the end of iteration 100 if they ask only the punisher, and the same agent society identifies the norm in iteration 7 if it an agent can ask any other agent for norm verification. As the probability of the other agent being a non-punishing agent is higher than being a punishing agent (0.96 vs. 0.04), the norm identification is faster when any agent that has a norm can recommend the norm to other agents (approach b) when compared to norm recommendation only by the sanctioning agents (approach a). In approach b, initially the norms are verified only by the punishers (i.e. the other agents do not know the norm). As more and more non-punishers come to know of the norm from punishers, they can also participate in the norm verification process. This leads to the faster convergence using approach b.

This simple experiment described in approach b reflects what happens in human societies where an agent asks and learns about a norm from another agent (Zastrow, 2009, pg. 192). Hence, a norm not only spreads just by the enforcer but also through agents that have been

---

\(^6\)We note that other mechanisms are also possible. For example, an agent can ask a certain number of agents in its vicinity instead of asking just one. In this work, agents ask one other agent for norm verification. There is a parameter in the system which can be used to change the number of agents that are asked for verifying a norm. This is akin to the referral process which has been studied by other researchers (Candale and Sen, 2005; Yu and Singh, 2002) which show that increasing the number of referrals leads to faster convergence.
Figure 7.4: Norm verification - enquiring about a norm to a punisher vs. any agent in the society

punished or an agent who knows about someone who has been punished. An agent entering a new society asks other agents in a society whether a suspected norm currently holds in a society.

We also conducted experiments with two types of punishers in the society, one that punishes eating activity and the other that punishes littering. The results were similar to the one shown in Figure 7.4 (i.e. asking any agent converges faster than asking just the punisher). Note that the experiments reported in the rest of the chapter use norm recommendations from any agent (approach b).

7.5.2 Experiment 2 - Dynamic norm change (from society’s view point)

Agents in a society should have the ability to infer norms when norms change. A new norm can be established when a punisher leaves the society and a new punisher joins the society or when a punisher agent comes up with a new norm replacing the old one. This experiment

This information can be obtained either through gossip (Paolucci et al., 2000; Boyd et al., 2006) or common knowledge (Chwe, 2001).
demonstrates that agents in different societies can change norms based on inference.

Figure 7.5: Simulation snapshot of the set-up for studying dynamic norm change

We experimented with four different societies. There were four punishers in total, out of which two agents punished littering actions, and two others punished eating actions in the park. These punishers were randomly assigned to different societies and the punishers are able to move from one society to another. The simulation set up is divided into 4 blue rectangles, and each rectangle represents a distinct society. The agents are represented as coloured circles (see Figure 7.5). The littering agents are in light blue, while the non-litterers are in light green. The punishers that punish littering actions are in red, and the punishers that punish the eating actions are in blue. A small rectangle that appears inside an agent represents the agent inferring the norm. The color of the small rectangle corresponds to the

---

8The simulation of this scenario can be viewed http://unitube.otago.ac.nz/view?m=3d8h11fu1xa. Note that the videos referred to in this thesis can also be found in the CD attached with this thesis.
punisher’s color (hence it corresponds to the identified norm). Figures 7.6 and 7.7 show the percentage of agents that have identified a norm in different societies as the punishers dynamically move across societies. The agents that punished litterers punished 25% of the punishing actions, while the agents that punished eating actions punished only 5% of the eating actions.

Figure 7.6: Norm change from the view point of a society (society 3)

Figure 7.6 shows how norms in society 3 (rectangle in the lower-left of the simulation window shown in Figure 7.6) change dynamically. When the simulation starts, there are both types of punishers in society 3, so agents start inferring both types of norms. About iteration 440, the punisher that punishes the eating action moves to society 4 (rectangle in the lower-right of the simulation window), and about iteration 463 the punisher that punishes littering action moves to society 4. As society 3 does not have any punisher, both the norms disappear from this society by iteration 560. The norms disappear because of two reasons. The first reason is that some agents with norms gradually move to other societies and the agents that come to the society may not know the norm to start with. In the absence of the punishers, these new agents do not infer the norms. This gradually reduces the number of agents with a norm in the society. The second reason is that some agents that know the
norms may re-evaluate the norm in certain intervals of time (by invoking the norm inference component) after the punishers have left. In the absence of the punishers these agent may remove the norm from their identified norm set. A punisher that punishes eating actions enters this society at iteration 944. Hence some agents start inferring this norm. Note that the new agents enter an agent society with a clean slate (i.e. they do not have any norms). That is the reason for the fluctuations of both the lines in Figure 7.6. If there are more new agents in the society that are without norms, the percentage of the agents with the norm will be small.

Figure 7.7: Norm change from the view point of a society (society 4)

Figure 7.7 shows how norms change in society 4. At the start of the simulation there is only one punisher in society 4, the punisher that punishes littering actions. So only one type is norm is inferred by the agents in the society. About iteration 164 this agent moves to society 2. Since there are no other agents to enforce the norm in the society, all the agents in this society gradually lose the norm by iteration 200 (for the same reasons presented in the previous paragraph). About iteration 440, a punisher that punishes eating action moves to this society, so agents start inferring the norm against eating in the park, and when the punisher that punishes littering enters in iteration 463, the agents start inferring the norm.
against littering. About iteration 500, another punisher that punishes the littering action enters society 4 from society 2. Note that at this point there are more punishers that punish littering actions than eating actions. This is a reason for more agents inferring the norm against littering than eating. By iteration 944, the punisher that punishes the eating action moves to society 3. Hence the norm against eating starts decreasing in the society.

In these experiments (Figures 7.6 and 7.7) we have not specified what the criteria for emergence is. We have only shown the percentage of agents that have identified the existence of a norm based on a sanction. Researchers have used different criteria for norm emergence in a society (35% to 100%, see Table 3.3).

7.5.3 Experiment 3 - Dynamic norm change (from individual agent’s view point)

An agent that moves across different societies may infer different type of norms. This experiment demonstrates that an agent is capable of inferring different norms in different societies using the norm identification architecture. Using the experimental set up described in section 7.5.2, we show how an agent can infer a norm (please see the video referred to in Section 7.5.2). In the video, it can be observed that there is an agent in black. This black agent is the observer that moves between societies 3 and 4. Using our architecture, this agent is able to infer different types of norms (a norm against eating, a norm against littering, and a norm against both eating and littering). Figure 7.8 shows the result of norm inference for an agent at different points of time. It can be observed that an agent is able to infer the change of norm when it moves from one society to another. At different points of time, the agent will have different norms.

![Figure 7.8: Norm change - an individual agent’s point of view](image)

143
From the simulation video, it can be observed that the agent in black is initially in society 3. It infers the norm against eating in that society (shown as a blue region in Figure 7.8). It moves to society 4 in iteration 64. As society 4 has a littering punisher, it infers that norm in iteration 115 (shown as a red region in Figure 7.8). The agent then moves into society 3 in iteration 177, where both type of punishers are present. It then identifies the norm against littering first and then the norm against eating\(^9\). Note that it identifies both types of norms (co-existing norms\(^{10}\)) in this society (regions with green diamonds in Figure 7.8) from iterations 250 to 268. As the agent moves in and out of societies 3 and 4, it identifies different norms depending upon the punisher(s) present in the society.

7.5.4 Experiment 4 - Adaptability of an agent

Norm Identification Frequency (NIF) and Norm Identification Threshold (NIT) are two parameters that an agent can vary based on its success in recognizing a norm in the society. The Norm Identification Frequency (NIF) refers to how often an agent invokes the norm inference component to obtain a candidate norm set. For example if the NIF is set to five, an agent invokes the norm inference components once in five iterations. The Norm Identification Threshold (NIT) refers to the threshold that an agent sets in order to obtain a candidate norm set. For example if NIT is set to 50% for an agent, the agent will seek a candidate norm set which contains those sub-episodes whose occurrence probability is greater than or equal to 50%.

7.5.4.1 Adapting Norm Inference Frequency (NIF)

An agent in our set-up starts with NIF=5. When it does not find any norm it retains the same NIF value. Once it has found all the norms in the society (i.e. there are no new norms to be found after a certain number of iterations), it increases its NIF value by a discrete amount (an increase of 5 in our case\(^{11}\)). The new value of NIF will be 10 which implies that the agent

\(^{9}\)This was because the probability of a punishment for littering was higher than eating.

\(^{10}\)In this work we have focused on simple co-existing norms (e.g. norms against eating and littering).

\(^{11}\)We have used a value of 5 in our simulations to model small fluctuations (increase or decrease) in the NIF value. This discrete amount is a parameter in our system which can be changed.
will infer norms only once in 10 iterations. This continues till a norm changes (e.g. a new norm is found). When a new norm is found, the NIF is set to back to 5. Figure 7.9 shows how the NIF changes by keeping NIT constant (50% in our case). It should be noted that when an agent moves into a new society, the NIF is set to 5.

![Adaptive Norm Inference Frequency (NIF) of an agent](image)

Figure 7.9: Adapting Norm Inference Frequency (NIF)

Figure 7.9 shows two lines corresponding to an agent moving in one society and within four societies with static punishers. All these societies have only one type of punisher (i.e. only one type of norm can be inferred). It can be observed that when the agent moves within one society, it infers the norm, and hence its NIF value increases (as the norm does not change). In the case of the agent moving across four different societies, the agent’s NIF increases after it has found a norm, as long as it is in the same society where it found the norm. When the agent moves to a new society, the agent’s NIF is set to the base value, and then it starts increasing once it has found the norm again. The “sudden jumps in values” (i.e. the vertical lines) that occur in regular intervals indicate that an agent has moved from one society to another.

---

12 We assume that the agent knows when it enters a new society. For example, the agent may know the physical boundaries of the society in which it is situated.
It should be noted that when the punishers are moving between societies (not shown in the figure), the NIF values (i.e. the NIF line) of an agent is different from when the punishers are static. In a set-up with static punishers, the punishers do not move (i.e. they remain in one society), but the other agents move. In a set-up with moving punishers, both the punishers and the other agents can move across societies. Hence, the NIF values for the same agent in these two set-ups are different. For example, let us assume that there is only one punisher in society 1. When the punisher moves from society 1 to 2, agents in society 1 are not able to recognize the norm when they try to infer the norm the next time (as dictated by their NIF value) as there are no sanctions. So, they may decrease their NIF value. On the other hand, in the set-up where punishers do not move, the agents in society 1 would have inferred the norm which would result in an increase in their NIF value.

7.5.4.2 Adapting Norm Inference Threshold (NIT)

An agent in our set-up starts with NIT=50%. When an agent invokes the norm inference component (NIF value is met), it initially uses the default NIT value. When the norms do not change the agent increases the NIT value by a discrete amount (an increase of 5 in our case). When no norm is found, it decreases the NIT value by a discrete amount (a decrease of 5 in our case). An agent increases its NIT because it can reduce extra computation that is needed to infer a candidate norm. For example if the same set of norms are obtained when an agent has NIT=100%, there is no reason why an agent should retain the value of NIT=50% where it has to perform some additional computation to infer the same norm (based on the PNI algorithm). An agent reduces the NIT in order to explore the search space of candidate norms below its initial threshold value. Note that when an agent moves into a new society, the NIT is set to 50%. Figure 7.10 shows how the NIT changes by keeping NIF constant (NIF=5 in our case). The vertical lines that occur periodically indicate that an agent has moved from one society to another. The initial value of NIT is set to 50%, because an agent should not have to start from scratch to infer a candidate norm (i.e. from NIT=0%) in order to avoid extra computation.

It can be noted from Figure 7.10 that when an agent moves within one society with one type of norm, its NIT threshold gradually reaches the value of 100, as it is able to infer the
norms with a high level of support. In the case of the agent moving across societies, an agent’s NIT value starts increasing when the agent has identified the norm, but it drops to 50 when it moves from one society to another. For the same set-up when the punishers are moving, the agent may not be able to infer the norms when the punishers move into a new zone. So, the NIT line showing the movement of an agent with static punishers is different from the scenario involving dynamic punishers (not shown in the figure).

7.5.4.3 Adapting both NIF and NIT

An agent can vary both NIF and NIT. Figure 7.11 shows how both of these variables change in an agent. The circled region is of interest, because it shows that an agent did not infer a norm initially (iteration 80 to 111). So the NIT line drops. It inferred the norm around iterations 111 to 133, which is indicated by the upwards trend. Then the agent did not infer the norm between iterations 134 to 139. Hence there is a drop in the NIT line. In iteration 140, the agent has moved to a new zone. The NIF line is similar to the one shown in Figure 7.9. The line that appears in the bottom shows when an agent moves from one society to another.
Experiments reported in sections 7.5.2, 7.5.3 and 7.5.4 have demonstrated that an agent can dynamically change its norm inference behaviour. An agent, being an autonomous entity, can decide when to increase or decrease these parameters to infer norms.

### 7.5.5 Experiment 5 - Using norm history

When an agent moves from one society to another, it can record the norm of the society it is leaving, so that the information can be used when it comes back to the same society. The experimental results shown in Figures 7.12 (a) and (b) demonstrate that the percentage of agents that have identified a particular norm in an agent society is high when agents store the norm (possess history) of the society. The recorded history can be used when an agent re-enters the society that it has previously been to. The graphs given in Figures 7.12 (a) and (b) have two lines each. The serrated red line shows the percentage of agents in a society with or without a norm. The straight line in green shows the average proportion of agents in the society with or without a norm.

Note that researchers have studied the impact of amount of history recorded by agents on
Figure 7.12: Comparison of the performance of the society with and without making use of norm history

convention emergence (see Section 3.2.3.2). The objective of this experiment is to compare how much better the agent will be if it records the norms of the society that it leaves. It can be seen that on average 77% of the agents can infer norms when history was stored (Figure 7.12 (b)) while only 47% of the other agents on average inferred norms (Figure 7.12 (a)). The experimental set-up was the same for both these experiments. A higher percentage of agents inferred norms when using their history, because the agents that come in with history information can start asking other agents in the society whether a norm holds (norm verification). If an agent does not have a norm history, it first has to infer the norm (i.e. invoke the norm inference component) and then ask another agent for norm verification, which is slower than using the norms in the norm history at the verification stage.

Using history is useful when the punishers are not moving (i.e. the norms in the society are stable). If the punishers are moving, then the norms may change (i.e. when there are different types of punishers). If the norms change, then the mechanism may not be very useful. If there are a large number of separate societies, then the agent may not come back to a previously inhabited society. In this case, the history information may not be useful. However, the agent is better off keeping the history if it comes back to a society it has previously been to, since it does not have to start inferring the norms from scratch.
7.6 Experiments based on the utility of an agent

An agent, being an autonomous entity, may choose to exercise its autonomy in order to maximize its utility. Such utilitarian agents may choose to become a part of a society that leads to an optimization of their utility. In this section we describe two experiments that we have conducted using utilitarian agents. The objectives of experiments are two-fold.

1. To demonstrate that the utility of a norm-abiding agent is better in a normative society where punishers are present than in a society where they are absent (i.e. a society where norm violations are not punished).

2. To demonstrate that when agents are capable of norm inference, the norm establishment in a society is faster than when agents do not infer norms.

7.6.1 Experimental set-up

Let us assume that there are two societies: a normative society and a society with no norms. There are three types of agents: learning litterers (LL), non-litterers (NL) and non-littering punishers (NLP) in both the societies. An agent has a utility value which we call the satisfaction level (S) which varies from 0 to 100.

An agent’s satisfaction level (S) decreases in the following situations:

- When a non-litterer observes a littering action its satisfaction level decreases (-1).
- When a litterer is punished, its utility decreases (-1).
- For all agents, littering activity results in the decrease of the utility. This is because each littering activity ruins the commons area (-1/number of agents in the society)$^{13}$.

An agent’s satisfaction level (S) increases (i.e. it gains utility) in the following situations:

- When a litterer litters, it gains utility in a society (+1).

---

$^{13}$Cleaning commons area such as a public park that has been littered would be at a cost which would be paid by the rate-payers of the city. From this point of view, bearing the cost of a litter has been distributed to all members of the society including the litterer.
• When a non-litterer does not see any littering action in a society, its utility increases (+1).

At the start of experiments, an agent moves across two societies (society 1 and 2). All the agents start with a satisfaction level of 50. This value can increase or decrease based on the society the agent is in. When the agents move into a new society, the satisfaction level is set to 50. When an agent has been to both the societies, an agent chooses the society for which it has a higher utility.

7.6.2 Experiment 1 - Norm abiding agents are better off in a normative society

Using the experimental set-up described in the previous section, we experimented how the agents organized themselves into two societies based on utility. In one of the societies, the punishers were present (society 2). In the other society the punishers were absent. In this experiment, the agents do not make use of the norm inference mechanism. They are utility maximizers (i.e. they would move to a society that yields better utility).

![Norm based separation in utilitarian society](image)

Figure 7.13: Separation of agents into two groups based on utility
At the start of the experiment, the agents moved in both societies. At regular intervals of time (every 50 iterations) each agent decides which society to choose. The agent evaluates its satisfaction level based on the societies it inhabited. The litterers’ utility in society 1 is better than in society 2, because in society 2 they are punished. So they move towards society 1. As the litterers move towards society 1, the non-litterers move towards society 2, because their utility in that society is better, due to the absence of litterers. It can be observed in Figure 7.13 that at the end 160 iterations, the litterers have moved from society 2 to society 1. When there are litterers in society 1, the non-litterers move to society 2. It can be observed that at the end of 350 iterations all the non-litterers have moved into society 2.

We also conducted experiments by making the punishers move across societies at certain intervals of time. In this set-up, we ran the experiments for 1500 iterations\textsuperscript{14}. In the first 500 iterations, the punishers were in society 2, and in the second 500 iterations the punishers moved to society 1. In the third 500 iterations the punishers moved back to society 2.

In the first 500 iterations, the agents were separated into two groups, as described above. After 500 iterations, when the punishers move to society 1, the utility of litterers in that society starts decreasing, so the litterers move to society 2. When they move to society 2, the non-litterers’ utility decreases, so they move to society 1. Again it was observed that the two societies were separated based on the personality of the agents (society 1 with non-litterers and society 2 with litterers). After 1000 iterations when the punishers have moved to society 2, the same process continues. Society 1 now has all the litterers and society 2 has non-litterers.

This experiment demonstrates that norm-abiding agents (i.e. non-litterers) are better off in the normative society, where the norm violation is punished by sanctioning agents, and the litterers are better off in a society with no norms. This experiment also demonstrates that a normative agent is adaptive, as it moves from one society to another if its utility decreases in the society\textsuperscript{15}.

\textsuperscript{14}Simulation can be viewed at http://unitube.otago.ac.nz/view?m=SE7M11esZnI. The litterers are in black, non-litterers are in green and the punishers are in red.\textsuperscript{15}This also applies to non-normative agents.
7.6.3 Experiment 2 - Utilitarian vs. Hybrid strategies for agents

In the previous experiment the agents employed the utilitarian strategy. In this experiment the agents use the norm inference mechanism. At the same time they also compute utility. We call this a hybrid strategy. Except for the change of the strategy the experimental set up was similar to the previous experiment. We observed that the overall separation of the two groups is faster when the agents are able to apply the norm-inference mechanism along with the utilitarian mechanism. This is because when the litterers in society 2 infer that there is a norm against littering, then they will decide not to litter in the society (to minimize the decrease in their utility) and also decide to move to another society. When a non-litterer in society 2 infers that there is a norm then it decides to stay in that society, as it knows that it would be better off in this society (as the punishers will punish the littering agents). So, the separation of agents into two societies is faster when a norm inference mechanism is used along with the utilitarian mechanism.

![Comparison of utilitarian vs. normative-utilitarian strategies](image)

Figure 7.14: Comparison of utilitarian vs. normative-utilitarian strategies

Figure 7.14 shows the number of litterers and non-litterers in two societies with the two types of strategies for the non-littering agents. It can be noted that when the hybrid strategy was employed by the agents, the separation of agents into two separate groups was faster.
than when the agents used just the utilitarian strategy. The hybrid strategy resulted in the littering agent moving from society 2 to society 1 faster than when using utilitarian society. In this experiment, the system using hybrid strategy converged 100 iterations earlier than the system that used utilitarian strategy. As the litterers moved to society 2, the non-litterers moved to society 1.

7.7 Summary

This chapter described how an agent can infer prohibition norms in a society using the norm identification architecture. The focus of this chapter was on the Prohibition Norm Inference (PNI) algorithm for identifying potential prohibition norms. The PNI algorithm discussed in this chapter makes use of a modified version of the association rule mining algorithm. We have shown that an adaptive agent can infer norms by varying different attributes of the system. Through simulations we have shown how the norm identification architecture allows for detection (i.e. identification), communication (i.e. verification) and modification (i.e. dynamic change of norms) of norms. We have also demonstrated that the norm inference mechanism is beneficial for an agent, as it learns about a norm faster than just using a utilitarian strategy.
Chapter 8

Identifying obligation norms

8.1 Introduction

In this chapter we present how an autonomous agent is able to identify obligation norms in a society using the Obligation Norm Inference (ONI) algorithm. Using agent-based simulation of a restaurant, we demonstrate how an agent can identify the tipping (gratuity) norm. We demonstrate that an agent that uses the norm identification mechanism is able to add, remove and modify norms dynamically. An agent can also flexibly modify the parameters of the system in order to identify the norms of the society.

The chapter is organized as follows. The restaurant scenario used in the identification of obligation norms is described in Section 8.2. Section 8.3 describes the Obligation Norm Identification (ONI) algorithm. Section 8.4 describes the simulation experiments that have been conducted and the results obtained. Section 8.5 provides a summary of this chapter.

8.2 Restaurant scenario

Let us assume that agents are situated in a restaurant in a virtual environment (e.g. Second Life). A new agent coming to a restaurant may not be aware of the protocol associated with ordering and paying for food items and the associated norms. For example, the protocol of a restaurant might be to first order and pay for the food before consuming the food, while the protocol of another restaurant may be that the agent can consume the food and pay the bill
at the end. The norms associated with the restaurants could be that the agent is expected to pay a tip after paying the bill or pay a tip along with the bill. This may vary from one culture to another. For example, in the USA one is expected to pay a tip while in New Zealand a tip is not expected. Depending upon the context the agent is in, failure to follow the norms may result in a sanction. In this chapter we explain how an agent can identify the norms and the protocols in a particular context (i.e. the restaurant scenario).

In the restaurant scenario, the agent is aware of the actions performed by an agent, which are the arrival of an agent (arrive), ordering an item (order), eating the ordered food (eat), paying for the ordered food (pay), tipping the waiter (tip), and departing the restaurant (depart). Let us assume that multiple agents come to the restaurant, consume food and move out of the restaurant.

Similar to what was discussed in Chapter 7, an observer agent would first observe the interactions that occur between the agents in the society. An event that is perceived by an agent consists of an event index, the agent(s) participating in that event, and the observed action(s). For example an agent observing an agent A arriving at the restaurant will represent this as $\text{happens}(1, \text{arrive}(A))$. This implies the observer believes that at time unit 1, agent A arrives at the restaurant.

A sample representation of events observed by an agent is given in Figure 8.1. There are four agents A, B, C and W in this scenario. A and B are customers while W is the waiter agent. Let us assume that agent C is the observer. Agent A arrives first and shortly afterwards agent B arrives. Agent A orders food with agent W. Agent A eats and then pays. Agent A then departs. Agent A is sanctioned by the waiter agent. Agent B orders food with agent W. Agent B eats and then pays. Agent B then departs.

An observer agent situated in an environment can sense actions of other agents through observation or through action logs that may be available. For example, in Massively Multi-Player Online Role Playing Games (MMORPGs), the logs of user interactions may be available for the observer through chat channels (cf. Boella et al. (2008)).

An agent records these events in its belief base. Event 7 is a sanctioning event, where agent W sanctions agent A. The reason for the sanction is that agent A failed to tip agent W. For an observer it may not be possible to know the reason for the sanction, unless it was specified a priori by the agent’s designer. In open agent societies, the norms of the society may not be known to an agent ahead of time. Additionally, the norms may evolve over
Figure 8.1: Sample event sequence

\[
\begin{align*}
\text{happens}(1, & \text{arrive}(A)) \\
\text{happens}(2, & \text{arrive}(B)) \\
\text{happens}(3, & \text{order}(A, W)) \\
\text{happens}(4, & \text{eat}(A)) \\
\text{happens}(5, & \text{pay}(A, W)) \\
\text{happens}(6, & \text{depart}, (A)) \\
\text{happens}(7, & \text{sanction}(W, A)) \\
\text{happens}(8, & \text{order}(B, W))
\end{align*}
\]

time. In order to infer a norm of the society, the agent will make use of the norm inference mechanism.

### 8.3 Obligation norm identification (ONI) algorithm

The objective of the obligation norm identification mechanism is to identify events that were expected to happen, but did not happen which results in a sanction. In the context of the restaurant scenario, an agent is expected to tip. When an agent does not tip, a sanction might occur. The ONI algorithm can be used to identify the reason for the sanction (i.e. the violation of the obligation norm of tipping). The details of the algorithm are presented in sub-sections 8.3.1 to 8.3.7.

#### 8.3.1 Parameters used in the algorithm

The parameters that are used in the Obligation Norm Inference algorithm are the same as the parameters described in Chapter 7. They are History Length (HL), Event Sequences (ES), Special Event Set (SES), Unique Event Set (UES), Occurrence Probability (OP), Window Size (WS), Norm Inference Threshold (NIT) and Norm Inference Frequence (NIF). In this algorithm, the Norm Inference Threshold (NIT) is used on two occasions (explained in Section 8.3.2). As the values of NIT for each of the occasions can be varied by an agent, there are two variables which are \(NIT_a\) and \(NIT_b\). Even though the parameters used in the
obligation norm identification process are similar to the identification of prohibition norms, the internal details of the algorithms are different. The ONI algorithm makes uses of the PNI algorithm twice to identify sub-episodes in the process of identifying obligation norms.

8.3.2 Overview of the algorithm

There are four main steps involved in the Obligation Norm Inference algorithm (see Algorithm 8.1). First, event episodes of a certain length are extracted from event sequences that an agent observes. These event episodes are stored in the Event Episode List (EEL). Second, based on the events in the special event set (e.g. sanctioning events), the event episodes in EEL are separated into two lists. The first list, called the Special Event Episode List (SEEL) contains all event episodes that contain at least one sanctioning event. The second list, called the Normal Event Episode List (NEEL) contains all event episodes that do not contain sanctioning events. Third, using SEEL, all sub-episodes which have occurrence probabilities greater than or equal to \( NIT_a \) are extracted and stored in Norm-Related Event Episode Set (NREES)\(^1\) using the PNI algorithm described in Section 7.3. Fourth, for each event episode in NREES, all supersequences are extracted from NEEL and stored in a temporary list called tempEEList. Based on the supersequences stored in tempEEList, the PNI algorithm can identify all permutations of supersequences with occurrence probabilities greater than or equal to \( NIT_b \). These are stored in Candidate Obligation Norm Set (CONS). These four steps are explained in detail in the following sub-sections.

\(^1\)Note that some of collections used are sets and lists. The sets are used to store unique event episodes, while the lists can hold duplicates.
Algorithm 8.1: Obligation Norm Inference algorithm (main algorithm)

begin
1 Create event episodes list (EEL);
2 Create special event episodes list (SEEL) and normal event episodes list (NEEL);
3 Extract norm-related event episodes set (NREES) from SEEL;
4 Create Candidate Obligation Norm Set (CONS) from NEEL using NREES;
5 /* Algorithm 8.2 */
end

8.3.3 Creating event episodes

An agent has a certain history length (HL). An agent at any point of time stores the history of observed interactions for the length equal to HL. A sample event sequence is given in Figure 8.1. When the norm inference component is invoked, the agent extracts, from the recorded history (i.e. event sequences (ES)) the events that involved a pair of agents. We call the retrieved event sequence the event episode (EE). A sample event episode from the viewpoint of an observer (agent C) is given below. The set on the left hand side of the colon indicates that the agents involved in the interaction are A and W. To the right of the colon is the event episode. A hyphen separates one event from the next.

\[
\{A, W\} : (happens(3, order, A, W) - happens(4, eat, A) - \\
happens(5, pay, A, W) - happens(6, depart, A) - happens(7, sanction, W, A))
\]

Using such observations, the observing agent may assume that something that agent A did in the past has caused the sanction. The focus of the previous chapter (Chapter 7) was on identifying prohibition norms where some key events that preceded the sanction were identified as the reason for the sanction. But, failure of agent A to perform certain action(s) could also have caused a sanction. The events that are absent are the obligations that are not fulfilled by the agents, and they are the reason for the sanction. Hence, the focus of this chapter is to identify those key events that are absent\(^2\). Agent C then extracts the sequence of events

\(^2\)The absent events are those missing events that were expected to happen, but did not happen.
(the event episode) that took place between A and W, based on the event sequence stored in its history. To simplify the notation, only the first letter of each event will be mentioned from here on (e.g. p for pay) and also the agent names are omitted. As the sequence itself caters for the temporal ordering of events, the event ids are omitted. Thus the event episode for interactions between agents A and W shown above will be represented as

$$\{A, W\} : (o - e - p - d - s)$$

Figure 8.2 shows a sample event episode list (EEL) that contains ten events involving a pair of agents that are observed by another agent where HL=6. Note that the Unique Event Set (UES) in this case includes events a,o,e,p,t,d,w,s which stand for arrive, order, eat, pay, tip, depart, wait, sanction respectively.

$$\begin{align*}
\{A, W\} &\rightarrow (e - p - d - s) \\
\{B, W\} &\rightarrow (e - p - t - d) \\
\{C, W\} &\rightarrow (a - o - e - p - t) \\
\{D, W\} &\rightarrow (a - o - e - p - d - s) \\
\{E, W\} &\rightarrow (p - d - s) \\
\{F, W\} &\rightarrow (o - e - p - d - s) \\
\{G, W\} &\rightarrow (a - o - e) \\
\{H, W\} &\rightarrow (a - o - e - p) \\
\{I, W\} &\rightarrow (a - d) \\
\{J, W\} &\rightarrow (a - o - e - p - t - d)
\end{align*}$$

Figure 8.2: Sample event episode list (EEL)

### 8.3.4 Creating special and normal event episode lists

Note that some event episodes in EEL have sanctions as one of the events. The agent identifies the sanction events from the special events set (SES). Using EEL, an agent creates two lists for further processing, one with event episodes that contain a sanctioning event and the other containing event episodes without sanctions. The list that contains event episodes with sanctioning events is called the special event episode list (SEEL). The other list is called the normal event episode list (NEEL).
The SEEL obtained from our example EEL is given in the left in Figure 8.3. NEEL has the remaining episodes that do not contain a sanctioning action (shown in the right of Figure 8.3).

Figure 8.3: SEEL on the left and the NEEL on the right

8.3.5 Generating the norm-related event set (NREES)

From the SEEL, an agent can identify events that have the potential to be associated with sanctions. For example, from the SEEL shown in the left of Figure 8.3, the agent may infer that the sub-episodes \( p-d \), \( p \), or \( d \) could be the reason for a sanction, as they occur in all the event episodes in SEEL. In the case of prohibition norms, the events that precede a sanction can be potentially linked to a sanction due to causality. In the case of obligation norms, it is the absence of an event or a sequence of events that might be the cause of the sanction. In both these types of norms, the agent has to identify the sequences of events that occur frequently before the occurrence of a sanctioning action. In the case of a prohibition norm, the frequency of occurrence may correlate with norm identification. In the case of an obligation norm, the agent first has to find the frequently occurring sequence(s), which are then stored in the norm-related event set (NREES). Let us refer to an event episode in NREES as \( \alpha \). Second, an agent has to identify all the supersequences of \( \alpha \) in NEEL with an occurrence probability greater than or equal to \( NIT_\alpha \), which are added to the candidate obligation norm set (CONS). The construction of NREES is discussed in this sub-section, and the construction of CONS is discussed in the next sub-section.

In order to identify these norm-related events, the agent uses the Prohibition Norm Identification (PNI) algorithm discussed in Chapter 7. The algorithm can identify candidate
norms that are obtained by considering “permutations with repetition” when constructing sub-episodes. Based on the SEEL, an agent can generate the NREES. Figure 8.4 shows the SEEL, on the left of the arrow, and the NREES generated from the SEEL on the right of the arrow when $NIT_a$ is set to 0. The occurrence probability of an event episode in NREES is given in square brackets. When $NIT_a$ is set to 0, all possible subsequences of event episodes in SEEL are generated. When $NIT_a$ is set to 100, the algorithm identifies the following norm-related event episode set \{p-d,p,d\}. An agent, being an autonomous entity, can vary the $NIT_a$ parameter to identify the norm-related events. Note that if an event episode is frequent, then all its subsequences are also frequent. For example if $p-d$ appears 100% of the time (i.e. the occurrence probability is 1), all its subsequences also appear 100% of the time.

$$\begin{pmatrix}
(e - p - d - s) \\
(a - o - e - p - d - s) \\
(o - e - p - d - s) \\
(p - d - s)
\end{pmatrix}
\rightarrow
\begin{pmatrix}
(p - d)[1] \\
(p)[1] \\
(d)[1] \\
(e - p - d)[.75] \\
(e - p)[.75] \\
(e)[.75] \\
(o - e - p - d)[.5] \\
(o - e - p)[.5] \\
(o - e)[.5] \\
(o)[.5]
\end{pmatrix}$$

Figure 8.4: SEEL on the left and NREES on the right

### 8.3.6 Identifying candidate obligation norm set (CONS)

The pseudocode for generating a candidate obligation norm set (CONS) is given in Algorithm 8.2. In order to identify the obligation norms, the agent has to identify those supersequences in NEEL that contain the event episodes in NREES. These supersequences are stored in a list ($tempEEList$ in this case).

Based on the supersequences stored in $tempEEList$, the Prohibition Norm Inference (PNI) algorithm can identify all permutations of supersequences whose occurrence probabilities are
greater than or equal to $NIT_b$. Such supersequences are stored in the candidate obligation norm set (CONS).

For example, let us suppose that the event episode $p-d$ is the only event episode stored in the NREES. Figure 8.5 shows the NEEL on the left of the arrow and the $tempEEList$ that is generated from the NEEL on the right. Note that the NEEL on the left contains six event episodes but $tempEEList$ contains two event episodes out of six that contain $p-d$. These two event episodes are supersequences of $p-d$.

**Algorithm 8.2**: Pseudocode to create the Candidate Obligation Norm Set (CONS)

**Input**: Norm-Related Event Episode Set (NREES), Normal Event Episode List (NEEL), Norm Identification Threshold ($NIT_b$)

**Output**: Candidate Obligation Norm Set (CONS)

1. \( CONS \leftarrow \emptyset \);
2. \textbf{for each event episode } \( NREE \in NREES \) \textbf{do}
3. \hspace{1em} \( tempEEList \leftarrow \emptyset \);
4. \hspace{2em} \textbf{foreach event episode } \( EE \in NEEL \) \textbf{do}
5. \hspace{3em} \textbf{if } \( EE \) is a supersequence of \( NREE \) \textbf{then}
6. \hspace{4em} Add \( EE \) to \( tempEEList \);
7. \hspace{3em} \textbf{end}
8. \hspace{2em} \textbf{end}
9. \hspace{1em} Use modified WINEPI algorithm to extract all candidate obligation norms (Input: \( tempEEList \), Unique Event Set (UES), Window Size(WS), Norm Inference Threshold ($NIT_b$), Output: Candidate norms);
10. \hspace{1em} Add candidate obligation norms to \( CONS \);
11. \textbf{return} \( CONS \);

From $tempEEList$ the CONS can be generated. The left hand side of Figure 8.6 shows the $tempEEList$. The right hand side of Figure 8.6 contains all permutations of supersequences of $p-d$ that can be obtained from $tempEEList$ and their occurrence probabilities in $tempEEList$ (in square brackets).
Assuming that $NIT_b$ is set to 100, the supersequences that will be identified as CONS are $p-t-d$ and $e-p-t-d$. Both these supersequences have an occurrence probabilities of 1.0. As the occurrence probabilities of $o-e-p-t-d$ and $a-o-e-p-t-d$ are less than $NIT_b$, these are not included in the CONS. Note that the PNI algorithm is used twice, the first time to obtain the NREES from the SEEL (Section 8.3.5) and the second time for obtaining the CONS from the NEEL using the NREES (line 9 of Algorithm 8.2).

For every event episode in the NREES, a new CONS is generated. Having compiled a set containing candidate obligation norms, the agent passes this information to the norm verification component to identify norms. This process is iterated until there are no elements in NREES. The norm verification process is explained in the next sub-section.

### 8.3.7 Norm verification

The norm verification mechanism is similar to what was discussed in Chapter 7. In order to find whether a candidate norm is a norm of the society, the agent asks another agent in its proximity. This happens periodically (e.g. once in every 10 iterations). An agent A can ask another agent B, by choosing the first candidate norm (say $p-t-d$ for which it has a higher
occurrence probability) and asks B if it knows whether the obligation norm $O_{X,Y}(t|p)$ is a norm of the society (i.e. an agent is obliged to tip after paying for the food ordered). If the response is affirmative, A stores this norm in its set of identified norms. If not, A moves on to the next candidate norm. In the case of the running example, the second candidate norm $e-p-t-d$ is chosen to be communicated to the other agent. It asks another agent (e.g. an agent who is the closest) whether it thinks that the given candidate norm is a norm of the society. If it responds positively, the agent infers $O_{X,Y}(t|(e-p))$ to be a norm. If the response is negative, the next norm in the candidate norm set is selected for verification. This process continues until a norm is found or no norm is found from the event episodes in the candidate norm set. Even in the case of successfully identifying a candidate norm, the agent continues the process to identify any co-existing norms.

Note that an agent will have two sets of norms: candidate norms and identified norms. Figure 8.7 shows the two sets of norms, the candidate norms on the left of the arrow and the identified norms on the right.

$$
\begin{pmatrix}
(p - t - d) \\
(e - p - t - d)
\end{pmatrix}
\rightarrow
\begin{pmatrix}
t
\end{pmatrix}
$$

Figure 8.7: Candidate norms on the left and the identified norm on the right

8.4 Experimental results

In this section we first describe the experimental set-up in sub-section 8.4.1. In the rest of the sub-sections we describe the experiments that were conducted and discuss the results obtained.

8.4.1 Experimental set-up

We model agents in our virtual society in a two-dimensional space. This virtual restaurant environment is shown in Figure 8.8. The agents can enter the restaurant and occupy one of the chairs at a table. Each table has six chairs. Each agent has a visibility threshold. The
visibility threshold of the agent is governed by a Chebyshev distance\(^3\) of length \(l\). An agent can observe actions of agents and the interactions that happen between two agents within its visibility threshold.

There are three types of agents in the simulation. They are non-tipping customers (NTC), tipping customers (TC) and waiters (W). There are eight possible types of actions defined in the simulation system: \textit{arrive, order, eat, pay, tip, depart, wait} and \textit{sanction}. The NTC agents can \textit{arrive, order, eat, pay} and \textit{depart}. The TC agents can \textit{arrive, order, eat, pay, tip} and \textit{tip}.

\(^3\)Chebyshev’s distance also known as the Chessboard distance is the minimum number of steps required for a King to move from one square of the chessboard to another. Chebyshev distance of length one corresponds to the Moore neighbourhood (Weisstein, 2010a) of size one where an agent in one cell can see all the 8 cells surrounding it.
and depart while the waiter agents can wait on the customers and sanction\textsuperscript{4}. The agents in this environment move from left to right. An agent chooses a seat and occupies it. It can then order food, eat and then pay for the food. The agent may tip the waiter. The agent may be sanctioned for not tipping. The agent can then depart the restaurant. The agents that are at the edge of the two dimensional space can again re-appear in the opposite side (i.e. a toroidal grid is implemented). The agents are represented as circles using different colours. The NTCs are red, the TCs are green, and the Ws are blue. The id of an agent and the action it currently performs appear above the circle. At any time step an agent can perform one action. When an agent does the same action over several steps, it is recorded as one action. For example if the agent eats for 10 iterations, the eating action is counted as one action\textsuperscript{5}. The same holds for the arrival and the departure of an agent. All the agents make use of the norm inference component to infer norms. The blue squares that appear within the circles represent the identification of a norm. The simulation parameters used in all the experiments are given in Table 8.1. A sample simulation can be viewed on the Web\textsuperscript{6}.

\textbf{8.4.2 Experiment 1 - Varying visibility threshold (V) and history length (HL)}

In this experiment there were 50 agents out of which 25 were tipping customers (TC), 21 were non-tipping customers (NTC), and 4 were waiter agents (W) who punished non-tipping customers probabilistically. The simulated environment was a 50*50 grid as shown in Figure 8.8.

An agent has a visibility threshold which dictates how many cells an agent can see. A visibility threshold of 5 would mean that the agent can see all agents which are at the maximum five cells away from it on all sides. The agent also has a stored history regarding

\textsuperscript{4}We note that the cost of punishment is not modelled in this work, because our main focus is to model and experiment with how an agent is able to recognize a norm in the first place. The cost of punishment has been investigated by other researchers (Fehr and Fischbacher, 2004; Ohtsuki et al., 2009).

\textsuperscript{5}We note this decision is domain specific. In some other domains such as an auction, it could be that an agent is prohibited from buying three consecutive items of the same type. In those cases each action of the same type should be recorded. We note that the mechanism proposed in this thesis can handle this scenario.

\textsuperscript{6}http://unitube.otago.ac.nz/view?m=wzig16WGY1p
Table 8.1: Simulation parameters for identifying the obligation norm of tipping

<table>
<thead>
<tr>
<th>Parameters</th>
<th>Values</th>
</tr>
</thead>
<tbody>
<tr>
<td>Grid size</td>
<td>50*50</td>
</tr>
<tr>
<td>Total number of agents</td>
<td>50</td>
</tr>
<tr>
<td>Number of tipping customers</td>
<td>25</td>
</tr>
<tr>
<td>Number of non-tipping customers</td>
<td>21</td>
</tr>
<tr>
<td>Number of waiters</td>
<td>4</td>
</tr>
<tr>
<td>Visibility threshold (V)</td>
<td>5 - 25</td>
</tr>
<tr>
<td>History Length (HL)</td>
<td>5 - 50</td>
</tr>
<tr>
<td>Norm inference threshold - a (NIT_a)</td>
<td>0 - 100</td>
</tr>
<tr>
<td>Norm inference threshold - b (NIT_b)</td>
<td>0 - 100</td>
</tr>
<tr>
<td>Window Size (WS)</td>
<td>1 - 10</td>
</tr>
<tr>
<td>Norm inference frequency (NIF)</td>
<td>5 - 50</td>
</tr>
<tr>
<td>Number of referrals (Ref)</td>
<td>1 - 10</td>
</tr>
</tbody>
</table>

the actions performed by other agents. When the history length is five, an agent stores the actions of all the agents within its vicinity in the last 5 iterations.

Figure 8.9 shows the rate at which a society identifies the obligation norm of tipping when the visibility and the history lengths of agents are varied. In all the graphs in Figure 8.9, the x-axis shows the iteration number and the y-axis shows the number of agents with the tipping norm.

By keeping the history length constant and all the other parameters constant we varied the visibility threshold for the agents. The top-left graph of Figure 8.9 shows the result of varying the visibility threshold. It can be noted that as the visibility of the agents increased, the agents identified the norms faster. This is because the agents were able to collect more evidence. This can be observed in all the graphs in Figure 8.9.

When the history length of an agent was increased the agents in the society inferred the

7An agent may initially set the visibility threshold to a lower value so that it does not have to process a large amount of information. If it does not find a norm, it can then choose to observe interactions that happen in a larger area by increasing its visibility.
norms faster. When we compare the results shown in the top-right graph of Figure 8.9 with the results shown in the top-left graph of Figure 8.9, it can be observed that as the history length increases the agents infer the norms faster. This can be observed as we move from the top-left graph to the bottom-right graph in Figure 8.9. When the history length is small, the event episode list may not contain all the information to infer a norm. But when the history length is increased, an agent will have better evidence to infer the norms.

8.4.3 Experiment 2 - Varying referral levels

Once an agent has identified a candidate norm, it asks other agents for norm identification within its visibility threshold. An agent can vary this parameter. It was noted that as the number of agents from whom an agent asked for referral increased (see Figure 8.10), the norm identification rate of the agents in the society increased\(^8\). The norm inference frequency in this experiment was once every 50 iterations. Other parameters of this experiment were:

---

8When the number of referees increases, the rate of norm establishment increases. This has also been reported in many other works in multi-agent systems(Yu and Singh, 2002; Yolum and Singh, 2003; Candale and Sen, 2005).
8.4.4 Experiment 3 - Varying Norm Inference Thresholds (NIT)

We have studied the effect of changing the NIT thresholds (a and b) on the size of NREES and CONS that are generated by an agent.

Figure 8.11 shows the size of the NREES when $NIT_a$ is varied. When $NIT_a$ is set low, the size of NREES generated by an agent is large. This means that an agent incurs a large amount of computation cost to generate NREES. When an agent sets $NIT_a$ high, the size of NREES is small. An agent, being an adaptive entity, can vary this parameter depending upon its success in identifying a norm.

We also conducted experiments to show the impact of varying both $NIT_a$ and $NIT_b$ on

$NIT_a=100$, $NIT_b = 80$, $V=25$, $HL=50$ and $WS=3$. 
the number of candidate norms generated (see Figure 8.12). When $NIT_a$ was set to 100 and $NIT_b$ was set to 100, a small set of candidate norms was generated (size = 9). When $NIT_a$ was set to 100 and $NIT_b$ was set to 50, 35 different candidate norms were generated. When $NIT_a$ was set to 50 and $NIT_b$ was set to 100, 37 candidate norms were generated. When $NIT_a$ was set to 50 and $NIT_b$ was set to 50, 57 candidate norms were generated. If an agent sets the NIT value high, the number of candidate norms that is generated is low. The number of candidate norms that are generated has an impact on the norm verification stage. The less the number of candidate norms, the less is the amount of time taken for norm verification. An agent can change these two parameters to adapt to the current situation. For example if an agent sets the NIT values to be high and it does not find a norm, it can decrease the value for the next norm inference instance.

8.4.5 Experiment 4 - Varying the Window Size (WS)

The objectives of this experiment are two-fold. They are to show that

- as WS increases, the accuracy of norm identification increases (i.e. the number of false positives decreases, and the number of candidate norms generated decreases)
- as WS is varied, different normative protocols associated with the norms can be iden-

---

9A norm inference instance is related to NIF. An agent infers a norm once every $x$ iterations as governed by NIF. When an agent invokes its norm inference component, this is known as a norm inference instance.
8.4.5.1 Accuracy of norm identification

By keeping other parameters constant we varied the Window Size (WS) of norm identification. The results of varying an agent’s WS for 20 norm inference instances is given in Figure 8.13. The success rate in identifying three different categories of norms are given in Figure 8.13. These three categories are 1) the only candidate norm identified is the tipping norm 2) tipping is identified as one of the candidate norms and 3) the tipping norm is not found and no norm is found.

When WS is set to 1, the agent identified the tipping norm to be the only candidate norm 5% of the time. 65% of the time, it identified the tipping norm as one of the candidate norms. In other words, the agent also had identified other candidate norms. These other candidate norms are false positives which are pruned during the norm verification stage. For the remaining 30% of the time, the agent either did not identify any norms or did not identify tipping as one of the norms.
As WS is increased, the accuracy of norm identification increases (i.e. the success of the agent in identifying tipping as the only norm increases). When WS=2, the agent identifies tipping to be the only norm 30% of the time. When WS=3, the agent identifies it 55% of the time and when WS is set to 5 it identifies it 85% of the time. It should be noted that as WS increases, the false positives decrease.

When WS is increased, the agent’s load in norm verification decreases (shown in Figure 8.14). When WS=1, about four candidate norms were generated in each norm inference instance. When WS=2, on average, more than two candidate norms were generated in each instance, and when WS is set to three, fewer than 2 norms were generated in each iteration. When WS was set to 5, only one norm was generated in each iteration. The standard deviation of the number of candidate norms identified is 1.08. It can be inferred that when the number of candidate norms generated is low, the amount of norm verification that needs to be done is low, which results in a reduction of communication between the agents.

There is no further improvement in the success of tipping norm identification when moving from WS=5 to WS=10, because the domain model supports a maximum of five different events that occur before a sanction (i.e. a-o-e-p-d). If the domain model were to be changed, the WS would need to be changed accordingly.
8.4.5.2 Identifying normative protocols

We define normative protocols to be the order of the occurrence of events (protocols) associated with a norm. For example, the protocol \( a-o-e-p-t-d \) defines the sequence that an average agent follows\(^{10}\). Normally, an agent arrives, occupies a seat and orders food, eats, pays, tips and then departs. The focus of this experiment is to demonstrate that an agent, by changing the WS, can have a partial or complete view of the what the normative protocol might be. For example, when WS is set to 1, the agent identifies only one event that precedes the sanctioning event. Hence, the size of the normative protocol identified will be two\(^{11}\). If WS is set to five, the size of the normative protocol that can be identified can vary from two to six\(^{12}\). Figure 8.15 shows the normative protocols generated by an agent over 20 norm inference instances.

From Figure 8.15, it can be observed that when WS is one, the agent identified \( PT \)\(^{13}\) as

\[ \text{Figure 8.15: Normative protocols generated by an agent for varying window sizes (WS)} \]

---

\(^{10}\)The protocol \( a-o-e-p-t-d \) includes the pre-conditions associated with a norm (i.e. \( a-o-e-p \)), the obliged action \( t \), and the post-condition associated with the norm (i.e. \( d \)).

\(^{11}\)For example, when WS=1, if \( p \) precedes a sanction, then \( p-t \) may be identified as the protocol. In the ONI algorithm, the size of the supersequence of a subsequence of size \( n \) is \( n+1 \).

\(^{12}\)When WS is set to 5, the length of the subsequences can vary from one to five. So, the length of the normative protocol can vary from two to six.

\(^{13}\)PT is \( p-t \).
the protocol, about 90% of the time. When WS was set to three, it identified, EPTD to be the normative protocol about 88% of the time. When WS was set to five and ten, EPTD and AOEPTD were the top two normative protocols that were identified. It should be noted that as WS increases, the size of the normative protocol identified increases. Also, the normative protocols identified for lower values of WS are subsets of the normative protocols identified using higher values of WS.

This experiment shows that an agent not only infers norms but also the associated normative protocol. An agent’s view of the normative protocol depends on the WS value of that agent. Note that, given a protocol, an agent can easily identify the norm associated with that protocol. For example, assuming that an agent knows the protocol in a restaurant is a-o-e-p-d (e.g. if this is given to the agent by the designer), an agent can easily identify a-o-e-p-t-d as the normative protocol based on observations of events. In our case, the normative protocol is inferred by the agent, without the protocol being given to the agent explicitly. The norm and the protocol are inferred through the norm identification mechanism. If the normative protocol were to be a-o-p-e-t-d in a restaurant (i.e. pay before eating and then tip), our approach would be able to identify the norm and the normative protocol. Further, the normative protocol can be split into preconditions and postconditions associated with a norm. For example, the precondition of the tipping norm could be a-o-p-e and the postcondition is to depart from the restaurant. The length of identified pre- and post-conditions are governed by the Window Size.

8.4.6 Experiment 5 - Adaptability of an agent in identifying norms

An agent in our system can flexibly modify the history length (HL) based on whether it is successful in identifying a norm. If an agent does not infer norms when HL=25, it can increase the HL. If it has identified that the norm holds at a particular HL, the agent will check after a certain number of iterations to determine whether the norm still holds. If it holds, it will try to decrease the HL and check whether the norm can be identified. If it can be identified, the agent will decrease its HL further. The objective for reducing the HL is to reduce the amount of computation required to find a norm. The agent will be better off in terms of the computation required if it can find the same norm when it lowers the amount of
Figure 8.16: Adaptive history of an agent

history it has to store.

The top line in Figure 8.16 shows the adaptive history length of an agent when it tries to identify a norm. The bottom line shows whether the agent has found a norm (a value of 5) or not (a value of 0). An agent initially starts with an HL of 25. When it does not find the norm when HL=25, it increases its HL by five. This increases to a maximum value of 50. Once a norm is found, the agent tries to check whether the same norm can be found for a lower value of HL. It can be inferred from Figure 8.16 that in iteration 75, the agent found the norm when the HL was set 40. When it tried to reduce HL to 35 (in order to reduce the amount of computation required to find the norm), the norm was not found. The agent then increases HL to 40 but does not find the norm due to changing environmental conditions (i.e. the amount of evidence collected with HL=40 did not result in finding the norm). Hence, the agent increased the HL to 45. Still, it did not find the norm. When it increased the HL to 50, it found the norm in iteration 250. It then decreased the HL to 45 and it found the norm again. When HL was set to 40 it did not find the norm, hence the agent increased HL to 45. This graph shows that an agent is adaptive in terms of the stored history (i.e. history length). Dynamic adjustment of history length will be beneficial to the agent when norms are changing.

We have also experimented with varying the history lengths of an agent with and without the ability of having an adaptive history length (i.e. static HL vs. dynamic HL). When HL is static, the agent has a constant HL throughout the simulation. When HL is dynamic, the
agent can change its HL based on whether it has identified a norm. It can be seen from Figure 8.17 that when dynamic HL is used, an agent is able to infer the norms faster. Note that when HL was set to five (static HL=5), the agent found the norm only after 99 norm inference instances. When a dynamic HL was used by the agent, it inferred the norm in 28 inference instances. For larger values of HL, there isn’t a significant difference between static and adaptive HL. This is because for large HL values the agent would have collected enough evidence from observing the other agents regarding the norm (i.e. the evidence of a sanction). For smaller HL values, the agent does not have enough evidence regarding the sanction. Hence dynamically adapting the history length produces better results for an agent.

Similar to varying the history length dynamically, an adaptive agent can also vary its visibility threshold, the number of referrals, norm inference thresholds, and the window size for identifying norms. The effects of changing these parameters have been reported in sections 8.4.2 to 8.4.5.

**8.4.7 Experiment 6 - Identifying dynamic norm change**

An agent should have the ability to dynamically add newly identified norms and remove norms that do not hold. This experiment demonstrates that norms can be added, removed and modified by an agent dynamically, depending upon the environmental conditions. The ability to change norms is important for an adaptive agent so that it can flexibly adopt norms.
An agent, on identifying a norm, evaluates whether the norm holds at regular intervals of time. If the norm does not hold, it removes the norm from its norm base.

Figure 8.18 shows the results of dynamic norm change in a society. There are two lines in the graph. The top line shows the proportion of agents in the society with a norm at any given iteration. The line that appears in the bottom shows whether punishers are present in the society. A (dummy) value of 5 means that there are punishers in the society, and a value of 0 means that the punishers are not present in the society. In this experiment, the punishers do not punish\(^{14}\) from iterations 300 to 600. Having found a norm, an agent checks for the validity of the norm once again after 50 iterations. If the norm is found again, then the agent does not delete the norm. If the norm is not found, it removes the norm from its norm base. When the punishers do not punish, the norm is not inferred. Since the norm is not inferred, the agent removes the norm. It can be observed that the agents start losing the norm from iteration 400, and all the agents in the society have successfully removed the norm by iteration 500. In iteration 700 some of the agents have identified the norm again, and all the agents have identified the norm in iteration 850\(^{15}\).

\(^{14}\)There can be several reasons why punishers may stop punishing. The punishers can move from one society to another, or they can just stop punishing because their utility has gone below a certain threshold.

\(^{15}\)The simulation video can be found at http://unitube.otago.ac.nz/view?m=ySYi16XA9s8
8.5 Summary

When compared to identifying prohibition norms, identifying obligation norms is difficult, because with a prohibition norm it is usually the occurrence of a particular event or a sequence of events that is the reason for a sanction to occur. With obligation norms, on the other hand, it is the absence of an event that is the cause of a sanction. Association rule mining algorithms only cater for the extraction of interesting sequences that are present in an event sequence. They cannot be directly used to identify sequences when data items are missing. To attend to this situation, the Obligation Norm Inference (ONI) algorithm presented here can be used to generate candidate obligation norms. Using a simple example (tipping norm identification in a restaurant), we have demonstrated how the norm inference mechanism works. In particular we have demonstrated that an agent can change the parameters of the system dynamically to infer norms. The agents can also add, remove, and modify norms based on the environmental conditions. We have also demonstrated that the mechanism can increase the accuracy of norm identification and reduce the number of false positives generated. The agents in the system can also identify normative protocols.
Chapter 9

Identifying conditional norms

9.1 Introduction

This chapter aims to answer the question of how agents infer conditional norms in a multi-agent society. Conditional norms are defined as norms with conditions. We thus distinguish norms that are not associated with conditions from the ones that have conditions. An example of a norm without a condition is the norm that prohibits anyone from littering a public park, i.e. \texttt{prohibit(litter)}. An example of a norm with condition is a norm that prohibits one from littering as long as there is a rubbish bin within \( x \) metres from the agent (e.g. \( \text{if (distanceFromBin < 10) then prohibit(litter)} \)). Software agents should not only have the ability to identify norms but also the conditions under which these norms hold.

Identifying conditional norms is important, because an agent that notices another agent getting punished when littering 25 metres from the bin may infer that the condition associated with the norm is the distance of 25 metres. But the actual norm could be that no one should litter within 50 metres from the bin. The utility of the agent can be negatively impacted through a sanction if it litters 30 metres away from a bin. In this case, the agent does not know the correct condition associated with the norm. Another example of a conditional norm is the tipping norm. In one society an agent may tip 10\% of the bill while in another society an agent might be obliged to tip 20\% of the bill. In this work we are interested in experimenting with the formation, modification, and the removal of conditional norms in the minds of the agents. We also show that identifying norms with conditions has an impact on
the utility of the agents.

Section 9.2 describes a mechanism for identifying conditional norms. Section 9.3 describes the experiments that we have conducted and the results obtained. A discussion on the conditional norm identification is provided in Section 9.4. The summary of this chapter is presented in Section 9.5.

9.2 Identifying conditional norms

In our framework when a new agent enters a society, it will try to identify the norms that currently hold in that society. Once an agent has identified a norm it may want to identify the context and the exact conditions under which the norm holds. In the park littering scenario, the context is the rubbish bin and the condition is the distance from the rubbish bin. When an agent identifies the norm in the first instance through observation, it may not know the exact conditions associated with the norm.

Let us assume that an agent upon identifying the norm knows the context of the norm\(^1\). For example on identifying that littering is prohibited, the agent identifies the presence of the bin as the context. The condition associated with the norm is the distance between the agent and the bin\(^2\). We call this a contextual condition.

Note that the condition associated with a norm will be specific to the domain under consideration. In the park littering example, the condition can be either one or two-dimensional. For example the distance between a littering agent and bin is a single dimensional entity. The littering zone can be modelled as a two dimensional entity if it is defined using \(x\) and \(y\) coordinates (i.e. an agent should not litter within 5 metres from bin’s \(x\) position and 10 metres from bin’s \(y\) position). Some researchers have used a two dimensional representation for normative conditions (Kollingbaum et al., 2008; Vasconcelos et al., 2009). In this work we have used the distance metric which we call the radius of the non-littering zone.

\(^1\)We assume that an agent knows the context based on the past experience or based on the domain knowledge. For example, an agent may know about littering from its past experience.

\(^2\)Proximity or the distance of interaction is a contextual condition in many social norms. For example, two people talking tend to speak in a low voice when they walk past others. Another example is the interpersonal distance norm (i.e. how close you can get to someone while talking without making him/her uncomfortable). Agents may be aware of the distance based contextual condition from their previous experience.
**Algorithm 9.1**: Pseudocode of an agent to identify conditional norm associated with the prohibition of littering

**Input**: Contextual Condition = distance from nearest rubbish bin

1. `maxDistanceFromBin ← 0, tempDistance ← 0; /* maxDistanceFromBin stores the value of the contextual condition */`
2. `conditionalNormReferralConsidered ← true; /* can be set to true or false */`
3. `conditionalNormRecommenders ← {};

4. **foreach** norm inference cycle **do**

    5. Obtain Norms Set (NS); /* By invoking Candidate Norm Identification algorithm */

    6. **if** `NS ≠ {}` **then**

        7. **foreach** norm in NS **do**

            8. **foreach** punished agent with the visibility threshold **do**

                9. `tempDistance ← getDistanceFromNearestBin;`

                10. **if** `tempDistance > maxDistanceFromBin` **then**

                    11. `maxDistanceFromBin ← tempDistance;`

                **end**

            **end**

        **if** `conditionalNormReferralConsidered` **then**

        14. `conditionalNormRecommenders ← getAgentsFromVicinity;`

        15. **foreach** agent ∈ conditionalNormRecommenders **do**

            16. **if** `agent.maxDistanceFromBin > maxDistanceFromBin` **then**

                17. `maxDistanceFromBin ← agent.maxDistanceFromBin;`

            **end**

        **end**

    **end**

**end**
Algorithm 9.1 shows how an agent identifies the conditional norm of the park. In each norm inference cycle an agent will first identify a set of norms using the norm identification framework discussed in Chapter 7. Let us assume that the agent has identified \textit{prohibit(litter)} as the norm which is stored in its Norms Set (NS). For each of the littering agents that were observed to be punished, an agent calculates the distance from the nearest bin to the punished agent using Chebyshev’s distance metric\(^3\). The agent finds the radius of the non-littering zone (lines 10-12) and stores it in \textit{maxDistanceFromBin}. The agent can choose to ask for referral from one or more agents (based on the variable \textit{conditionalNormReferralConsidered} in line 2) from its vicinity threshold regarding the zone in which littering is prohibited (i.e. \textit{maxDistanceFromBin}). If the referrer’s recommended distance is greater than distance observed by the agent, the agent increases the distance (lines 14-21).

While Algorithm 9.1 is specific to the park littering scenario, the generic process of an agent to identify the conditional norm is given in Algorithm 9.2. Once the agent infers a norm, it will identify the contextual condition. The contextual condition can contain multi-dimensional attributes. For each norm in the norm set (NS), it calculates the value of the contextual condition (line 8). An agent calculates the value for contextual condition based on observing all the punished agents within its visibility threshold.

\(^3\)In our implementation Chebyshev distance represents the minimum distance between an agent and the nearest bin.
Algorithm 9.2: Pseudocode of an agent to identify a conditional norm

Input: Contextual Conditions

valueOfContextualCondition[] ← ∅;
conditionalNormReferralConsidered ← true;
conditionalNormRecommenders ← ∅;

foreach norm inference cycle do

Obtain Norms Set (NS); /* By invoking Candidate Norm Identification algorithm */

if NS ≠ ∅ then

foreach norm n in NS do

valueOfContextualCondition [n] ←
calculateContextualConditionalValue; /* This is calculated based on the available data on all punished agents within the visibility threshold */

if conditionalNormReferralConsidered then

conditionalNormRecommenders ← getAgentsFromVicinity;

foreach agent ∈ conditionalNormRecommenders do

if agent.valueOfContextualCondition is better than valueOfContextualCondition then

valueOfContextualCondition ←
agent.valueOfContextualCondition;

end

end

end

end

end

The observer agent can optionally ask recommendation from other agents (through refer-
ral), on the contextual condition that they have observed (lines 9 and 10). Then, based on the recommendation of other agents it can choose the best value as its value for the contextual condition (lines 11-15).

### 9.3 Experiments

In this section we first describe the experimental set-up in sub-section 9.3.1. In the rest of the sub-sections, we describe the experiments that were conducted and the results obtained.

---

*Figure 9.1: Snapshot of the simulation of conditional norms*

---

4The logic for choosing the best value of the condition is domain specific. In a domain the best value may correspond to the least numeric value and in another domain it may correspond to the highest value.
9.3.1 Experimental set-up

The experimental set-up for this experiment is similar to the set-up presented in Chapter 7. We model agents in the virtual society in a two-dimensional space. This virtual environment can be considered as a communal region such as a park shown in Figure 9.1. The agents explore and enjoy the park by moving around. There are three types of agents in the simulation. They are learning litterers (LL), non-litterers (NL) and non-littering punishers (NLP). There are four possible types of actions defined in the simulation system: move, eat, litter and punish. The LL agents can move, eat and litter. The NL agents can move and eat while the NLP agents can move, eat and punish. The agents’ movement can be in one of the four directions: up, down, left or right. The agents that are at the edge of the two dimensional space can again re-appear in the opposite side (i.e. a toroidal grid is implemented). The agents are represented as circles using different colours. The LLs are green, the NLs are blue and the NLPs are red. The identification number of an agent and action it currently does appear above the circle.

Each agent has a visibility threshold. The visibility threshold of the agent is governed by a Chebyshev distance of a certain length. An agent can observe actions of other agents, and the interactions that happen between two agents within its visibility threshold. The dashed square that appears at the bottom of Figure 9.1 shows the visibility range of agent 13 which is at the centre of the dashed square with a Chebyshev distance of four. All the agents make use of the norm inference component to infer norms. The red squares that appear within the circles represent the identification of a norm. Rubbish bins in the simulation environment appear in orange. The non-littering zone with reference to the bin at the top is given by the dashed square that appears at the top of Figure 9.1. The radius of non-littering zone in this case is four.

As an agent moves around a society, it knows the absolute positions of the rubbish bins. It knows its current position in the grid and the positions of all the agents within its visibility threshold. Using the knowledge of the positions of the agents, it calculates the distance between the sanctioned agents and the nearest bin.

The simulation parameters that were kept constant for all the experiments are given in
Table 9.1: Simulation parameters for identifying conditional norms

<table>
<thead>
<tr>
<th>Parameters</th>
<th>Values</th>
</tr>
</thead>
<tbody>
<tr>
<td>Grid size</td>
<td>20x20</td>
</tr>
<tr>
<td>Total number of agents</td>
<td>20</td>
</tr>
<tr>
<td>Number of litterers</td>
<td>12</td>
</tr>
<tr>
<td>Number of non-litterers</td>
<td>4</td>
</tr>
<tr>
<td>Number of non-littering punishers</td>
<td>4</td>
</tr>
<tr>
<td>Visibility threshold</td>
<td>5</td>
</tr>
<tr>
<td>Number of rubbish bins</td>
<td>2</td>
</tr>
<tr>
<td>Radius of non-littering zone (maxDistanceFromBin)</td>
<td>10</td>
</tr>
<tr>
<td>Number of referrals (when used)</td>
<td>1</td>
</tr>
</tbody>
</table>

Table 9.1. A sample simulation can be viewed from UniTube\(^5\).

### 9.3.2 Experiment 1 - Conditional norm identification

The objective of the first experiment is to show that agents in a society infer conditional norms using the proposed mechanism. We also compare the rate of norm establishment in the society with the rate of conditional norm establishment in the society.

Figure 9.2 shows two lines that represent the proportion of agents with norms and the proportion of agents with conditional norms in a society respectively. It can be seen from Figure 9.2 that even though the norm has been established in the society\(^6\) in iteration 270, the conditional norm (i.e. the agent should not litter when it is within 10 metres from the bin), is not inferred in the society till iteration number 410. This is because the conditional norm identification process is invoked by an agent after it has found a norm. As the agents interact more and more in the society, they gather more evidence regarding the condition associated with the norm. If the norm does not change, then the correct condition associated with the norm is inferred eventually. When an agent does not infer a norm for a certain amount of

---

\(^5\)http://unitube.otago.ac.nz/view?m=iWs217vmY6H

\(^6\)We assume that a norm is established in a society if all the agents (100%) have inferred the norm. Researchers have used different criteria ranging from 35% to 100%.
time or when the norm changes, it will remove the norm and its associated conditions from its norm base.

9.3.3 Experiment 2 - Conditional norm identification with and without referral

An agent can expedite the process of identifying a conditional norm if it asks another agent for its evidence of the normative condition. We refer to this as the conditional norm referral process. It can be observed from Figure 9.3 that when the referral is used, the rate of establishment of the conditional norm increases. The agents ask for referral from one other agent in the society. When the number of referees increases, the rate of conditional norm establishment increases (not shown here, but has been shown in another context in Figure 8.10). This has also been reported in other works in multi-agent systems (Yolum and Singh, 2003; Candale and Sen, 2005).

Figure 9.4 shows the progression of two agents towards the identification of the correct conditional norm (non-littering zone of radius 10) with and without referrals. The progression rates of the two agents are different because of their different paths of travel. If an agent observes more agents on its path, then it has a higher probability of inferring both the norm
Figure 9.3: Rate of conditional norm establishment in an agent society with and without referrals

and the condition associated with the norm. It should be noted that the conditional norm establishment for these two agents improve when the referrals are used.

The two dashed lines in Figure 9.4 show the radius of the non-littering zone identified by the agents during the simulation. The agent which found the norm first (agent 1, iteration 90) was not the one to find the correct conditional norm first\(^7\). When agent 1 found the norm in iteration 90, the non-littering zone identified by the agent was 6 metres (shown using an arrow in the Figure). It found the correct conditional norm in iteration 380. Agent 2, albeit finding the norm second (iteration 110, non-littering zone of radius 7 metres), found the correct conditional norm faster (iteration 190). This again is governed by the number of agents an agent gets to observe (i.e. the path of travel).

The two solid lines show how the radius of the non-littering zone was continually identified by the agents when referrals were used. It is interesting to note that when the referral mechanism was used, the agent which found the norm first was also the one that found the normative condition first. This is because once the agent finds the norm, it can ask the agents in the vicinity for referrals instead of waiting for a long amount of time to find out the maxi-

\(^7\)The correct conditional norm is the non-littering zone of 10 metres.
9.3.4 Experiment 3 - Dynamic conditional norm identification

An agent should have the ability to dynamically add newly identified norms and remove norms that do not hold. This experiment demonstrates that conditional norms can be added, removed and modified by an agent dynamically depending upon the environmental conditions. The ability to change norms is important for an adaptive agent so that it can flexibly adopt norms. An agent, on identifying a norm, evaluates whether the norm holds at regular intervals of time. If the norm does not hold, it removes the norm from its norm base. When it removes the norm, it also removes the condition associated with the norm.

Figure 9.5 demonstrates that an agent is able to add and remove norms and normative conditions dynamically. Figure 9.6 demonstrates that agents in our system are able to dynamically identify the modified normative conditions. In these two experiments, the punishers do not punish from iterations 1000 to 1250. This is to simulate the change in the environment which triggers a norm change. Additionally, having identified a norm, an agent checks for the validity of the norm once again after 5 norm inference cycles (norm inference happens once every 10 iterations). If the norm is found again, then the agent does not delete the norm. If the norm is not found, it removes the norm and the conditions from its norm base.
Figure 9.5: Dynamic conditional norm identification by an agent

Figure 9.6: Dynamic conditional norm identification by an agent
Figure 9.5 shows two lines that represent an agent adding and removing norms based on changing environmental conditions. The red line represents the agent using the referral mechanism and the blue line represents the agent without using the referral mechanism. It can be observed that the agent not using referrals identifies a conditional norm in iteration 60, and the correct conditional norm in iteration 120 while it inferred the norm faster when it used referrals. In this experiment, when the punishers do not punish, the norm was not inferred for 50 iterations (5 norm inference cycles starting from iteration 1010 to 1050). So, the agent removed the norm and the conditions associated with the norm (with and without referral) in iteration 1050. The agent that did not use the referral found a conditional norm again in iteration 1280 and the correct conditional norm in iteration 1670. It can be observed that when the referral was used by the agent, it identified the correct conditional norm earlier (iteration 1330).

Figure 9.6 shows two lines that represent the identification of different normative conditions under changing environmental conditions (with and without change of non-littering zone) for the same agent. By keeping all the other parameters the same, we varied the radius of the non-littering zone (i.e. the punishment zone for littering). This is to demonstrate that when the radius of the littering zone varies, the agent infers the change. After iteration 1250 all NLP agents punished only those agents that littered within 5 metres from the bin (as opposed to 10 metres which was used in iterations 1 to 1000). It can be observed from the green line in Figure 9.6 that the agent inferred the new normative condition (radius = 5). Note that the agent has made use of referrals in this experiment. The red line which converges to the littering zone of radius 10 (at the top of Figure 9.6) is the same as the red line shown at the top of Figure 9.5, which represents the normative behaviour of an agent that uses the referral process.

The simulation video can be found at http://unitube.otago.ac.nz/view?m=nQ6y17rCcJ
9.3.5 Experiment 4 - Comparison of utility of agents with and without conditional norm identification

The objective of this experiment is to compare the utility benefits of an agent when it identifies a norm with and without conditions. In this experiment, an agent has a utility value which we call the satisfaction level (S) which varies from 0 to 100.

An agent's satisfaction level (S) decreases in the following situations:

- When a litterer is punished, its utility decreases (-1).
- For all agents, littering activity results in the decrease of the utility. This is because each littering activity ruins the commons area (-1/number of agents in the society).

An agent’s satisfaction level (S) increases (i.e. it gains utility) in the following situation:

- When a litterer litters, it gains utility in a society (+1).

We have experimented with the utility of the agent with and without conditional norm identification. An LL agent is better off by using conditional norm (CN) identification. Having identified a norm an LL agent may choose to abstain from the action that is being prohibited. In the case of a conditional norm, it learns the exact condition under which it should not violate the norm. By this process, it can improve its utility. It can be observed from Figure 9.7 that an LL agent’s utility increases (greater than 50) when it has identified the conditional norm than just identifying the norm without conditions (less than 50). This is because when a littering agent finds the norm without the normative condition, it entirely abstains from the littering activity which does not lead to an increase in its utility. But when it identifies the normative condition, it now can litter outside the non-littering zone, which results in an increase in its utility.

For an NL agent, when it identified a norm without identifying the condition, the utility initially decreases but then stabilizes to a constant value because when all the agents inferred the norm, there aren’t any littering actions in the society. When agents in a society are capable of identifying conditional norms, an NL agent’s utility continues to decrease, because whenever an LL agent litters outside the not-to-litter zone, it leads to the decrease in the utility of the NL agent (because of sharing the littering cost with the entire society). Even
though an LL agent also incurs sharing the cost of littering of others, its net utility increases as it gains from its littering action (i.e. littering outside the non-littering zone).

It should be noted that when the utility of an NL agent goes below a certain threshold, it can leave the society, or can become a litterer or a punisher. Additionally, if the parameters of this experiment are varied, for example if the utility gain of a litterer is changed to 0.5 and the utility loss on receiving a punishment is changed to 0.25, the results obtained will be different. The objective here is to show that the conditional norm identification has an impact on the utility of the agents.

The utilities of NLP agents are not discussed here because we assume these agents have other utility functions for punishing (e.g. a leader who wants to promote a smoother functioning of the society, or an altruistic agent who does not care about its diminishing utility). We note that if non-altruistic punishers are present in the society, then the cost incurred by the non-altruistic punishers will play a role in the establishment of a norm in the society. Several research works have investigated the role of punishment costs on norm spreading (Ohtsuki et al., 2009; Fehr and Fischbacher, 2004).

Figure 9.7: Utility comparison of two agents
9.4 Discussion

The issue of conditional norm identification has not been dealt with by researchers in the field of normative multi-agent systems. To this end, we have investigated how a conditional norm can be identified by an agent in the context of park-littering. Identifying norms with conditions can be beneficial in several settings. For example, the norm identification architecture can be used to infer norms in Massively Multi-player Online Games (MMOGs). Players involved in massively multi-player games perform actions in an environment to achieve a goal. They may play as individuals or in groups. When playing a cooperation game (e.g. players forming groups to slay a dragon), individual players may be able to observe norms. For example, a dragon can only be slain if two players are within certain distance from the dragon. An agent that identifies this condition (the distance) will be better-off than an agent that just infers the norm of cooperation (i.e. two players are needed to slay a dragon). The mechanism proposed in this chapter can be used to identify norms with conditions.

Another application of identifying conditional norms is in the area of e-commerce. For example, in one society, the norm associated with the deadline for payment (i.e. obligations with deadlines as in Cardoso and Oliveira (2010)) may be set to 120 minutes after winning the item. Depending upon what an agent has observed, agents may have subtly different norms (e.g. one agent may notice that pay follows win after an average of 80 minutes while another may notice this to happen after 100 minutes). Both these agents could still infer the norm but the deadlines they had noticed can be different. This may result in an unstable equilibrium with reference to the norms and hence conflict resolution mechanisms should be used to resolve them (Kollingbaum et al., 2008; Vasconcelos et al., 2009).

We note that we have modelled and experimented with a simple domain. The number and type of agents can easily be increased and the normative conditions identified can be richer and more complex depending upon the problem domain. The main contribution is the mechanism for the identification of conditions associated with norms.
9.5 Summary

This chapter addressed the question of how conditional norms can be identified in an agent society using the norm inference architecture. Identification of conditional norms has been demonstrated in the context of a simple park-littering scenario. The ability of an agent to add, delete and modify a conditional norm has also been demonstrated. It has also been shown that identifying norms with conditions has an impact on the utility of the agents in the society.
Chapter 10

Discussion

The objectives of this chapter are two-fold. First, we discuss the contributions of this thesis. Second, we discuss the limitations and the future research directions of the norm identification aspect of this thesis.

10.1 Contributions of this thesis

The main contributions of this thesis are to two sub-areas in the field of normative multi-agent systems (NorMAS), namely norm emergence and norm identification. It also contributes to the field in general through the categorization (or taxonomy) of mechanisms investigated by the empirical works on norms and the life-cycle model of norms that has been proposed.

10.1.1 Norm emergence

The thesis makes two contributions to the area of norm emergence. First, a role-model based mechanism for norm emergence has been proposed and investigated (Chapter 4). Most previous work on norms has used a centralized leadership mechanism for norm spreading and emergence. The role model mechanism discussed in Chapter 4 uses the concept of distributed normative advice, whereby a hierarchy of leadership is achieved (a local leader may have several followers and can follow another leader). The norms emerge on top of the

---

1The limitations and future research directions in the area of norm emergence are discussed in Chapters 4 and 5 respectively.
leadership network.

Second, earlier work on norms has investigated how norms emerge on top of static network topologies. In Chapter 5 we have demonstrated how norms emerge on top of dynamically changing network topologies using the role model mechanism for norm emergence. We have also demonstrated the use of a particle-collision mechanism to model the bringing together of two different societies and have investigated how norms emerge in the newly formed society.

10.1.2 Norm identification

This thesis makes several contributions to the area of norm identification. The main contribution in this area is the initial investigations into answering the question of “how an agent can identify a norm in an open agent society”. To that end, we have proposed an internal agent architecture to identify norms in open agent societies (Chapter 6). The architecture makes use of signalling as the top level construct to identify both sanctions and rewards. Based on the occurrence of these signals, an agent can identify two types of norms, the prohibition norms (Chapter 7) and obligation norms (Chapter 8) by employing an association rule mining approach.

The thesis also demonstrates the following:

1. An adaptive agent can infer norms based on its success in identifying norms (by changing different parameters of the system if it is not successful). For example, an agent can modify the norm inference threshold and norm inference frequency parameters to infer potential norms.

2. An agent can dynamically add, remove and modify norms. An agent in our framework can add an identified norm to its norm set and also remove the norm when it does not hold.

3. An agent can identify co-existing norms. Agents employing our framework can infer co-existing norms in a park such as a norm against littering and a norm against eating.

4. An agent can identify conditional norms. Agents in our system can identify conditions associated with norms. For example, the agents can infer the distance from the rubbish...
bin within which agents are expected not to litter.

5. An agent can identify normative pre-conditions. Not only can an agent identify the norm, but also can infer the precondition associated with a norm. For example, in the context of the restaurant scenario, an agent may identify that actions eat and pay usually precede the tipping action.

10.1.3 Developmental phases of norms and the taxonomy of empirical works on norms

We have proposed a life-cycle model for norms in Chapter 3. We have identified five developmental phases of norms namely creation, identification, spreading, enforcement and emergence. We have explained the relationships between these phases. For selected works on norms, we have shown what phases of the life-cycle model have been taken into account in these works.

In this thesis we have also identified nine different mechanisms investigated by empirical works on norms (Chapter 3). We have also discussed how these identified mechanisms are employed in different developmental phases of the norm life-cycle model.

Figure 10.1 shows the contributions of the thesis (highlighted in blue text) to five categories (or mechanisms) as discussed in Chapter 3. They are the following:

1. investigation of the distributed role-model based leadership mechanism for norm emergence (Chapters 4 and 5)

2. consideration of network topologies (static and dynamic network topologies investigated in Chapters 4 and 5 respectively)

3. proposal of the internal agent architecture for a cognitive agent to infer norms (Chapters 6 to 9)

4. use of association rule mining (a data mining approach) to infer norms (Chapters 6 to 9)

5. use of a sanctioning mechanism to punish agents (Chapters 6 to 9)

\(^2\)Figure 10.1 is the same as Figure 3.5.
10.2 Limitations and future research directions

In this section we first describe some of the limitations of our current work on norm identification. When describing a limitation, we also provide pointers to how this can be addressed. Second, we describe the future research directions.

10.2.1 Limitations

1. Only observable events are identified as norms - In our work, only observable events can be identified as norms. For example, littering a park is observable and also not pay-
ing a tip is observable. On the other hand, someone might receive a sanction two days after they had done something wrong (e.g. receiving a speeding ticket after two days). In this case, the event that caused the sanction may not necessarily be observed since an agent may not store a very long history. In those cases, the present approach will not work\(^3\). Our work will suit social norms where the sanction is immediate for the events that are observable. We believe most social norms have immediate sanctions. However, sanctioning mechanisms such as the lowering of the reputation may not be directly observable as the action is internal to an agent. Our approach works for sanctioning mechanisms that have explicitly visible and interpretable actions (e.g. gestures such as shaking head in disapproval and yelling). Note that an assumption of our work is that emotion-based actions can be interpreted and categorized into sanctions and rewards. Several works have dealt with recognizing emotions in virtual societies (Picard, 1998; Neviarouskaya et al., 2009; Koda and Ruttkay, 2009). A component that categorizes events can be integrated with the framework in the future.

2. **Simple simulation models** - In this thesis, we have investigated norm identification based on simple experimental models. These models defined simple roles for agents. For example, the park scenario the agents had three roles: litterer, punisher and non-litterer. The restaurant scenario had roles such as waiter, punisher and customer. We believe our work can be extended to include multiple roles for the same agent and also include hierarchies of roles. Additionally, the number of actions considered in the simple domains are small (four and eight in chapters 7 and 8 respectively). In the future richer and complex domains can be considered.

### 10.2.2 Future research directions

1. **Emergence of signals (sanctions or rewards)** - Our current work assumes that a punisher agent knows a priori the action that should be sanctioned. Though this may hold

\(^3\)This is a granularity issue. An agent can choose to record all the events or only some of the key events that happen around it. By choosing to record key events, the agent may have a coarser but longer history which can be used to infer norms. Additionally, if sanction follows long after the action has been performed, it is difficult to associate a sanction with the action that triggered it (i.e. it is computationally expensive).
for norm leader or entrepreneur agents that come up with norms, in some scenarios, the action that is sanctioned may not be known even to the potential sanctioning agent ahead of time. The sanction might emerge depending upon the environmental dynamics. For example, an agent might not sanction if it sees one agent littering. But, when it sees $n$ agents littering the park, it might start punishing, because that action has lowered its utility beyond a certain threshold (an internal utility function). In this scenario, an agent can use a learning algorithm (e.g. Q-Learning (Watkins and Dayan, 1992)) to identify an action that lowers its utility and then can sanction that action. The norm then can be identified from the sanction that has emerged. The extension here would be to include a mechanism for the emergence of sanctions and then apply the mechanisms proposed in this thesis to identify the norm.

2. **Alternative mechanisms to norm verification** - An agent can verify whether its candidate norms hold by undertaking actions that it observes to be sanctioned (e.g. by violating the tipping norm). Based on the outcome of tests the agent carries out, it could infer what the norms could be. This would be a meta-level norm-testing mechanism of an agent. This would also solve the lying problem (i.e. a norm verifying agent may potentially lie about whether a norm holds in a particular society).

Another alternative to norm verification is to include humans in the loop. The agents that are proxies to humans can recommend candidate norms to humans and the humans can then decide which ones are norms. Agents in this case play the role of a norm recommender. Such a norm recommender system can recommend both the norm (e.g $prohibit(litter)$ might be the norm which is inferred based on an agent identifying the fact that whenever there is a sanction, littering action is involved 100% of the time) and the norm strength (e.g. the frequency of punishing the littering activity is 50%). These extended mechanisms can be explored in the future.

3. **Integration of our framework with other research works** - We believe our work on norm identification can be integrated with the other research work whose focus lies in agent decision-making concerning whether to follow a norm. A brief outline on how this might be realized is given below.
Once a norm has been identified by an agent, the agent then has to decide whether it should follow the norm. There are several architectures that have been proposed that an agent can use to decide whether to follow the norm, such as the extended BDI architecture by Meneguzzi and Luck (2009) as well as a personality based approach by López y López et al. (2002) and its extension (Criado et al., 2010).

Figure 10.2 shows how our research work on norm identification can be integrated with some of the existing work in this area. The figure shows the extended framework that an agent can use to identify norms and also decide whether to follow the norm. The dashed rectangle in Figure 10.2 represents other existing works. We call the component that assesses the applicability of a norm in a particular situation the norm applicability assessment component (the rounded solid rectangle that appears inside the dashed rectangle in Figure 10.2).

An agent may invoke the norm applicability assessment component under two circumstances. First, when an event occurs it can check if it violates its p-norms or g-norms. The agent then invokes the norm applicability assessment component to decide whether a norm applies and what action it should take when the perceived event is against the norm. This decision may depend upon several factors such as an agent’s utility (e.g. personality (López y López et al., 2002), goals, desires and intentions (Meneguzzi and Luck, 2009)). When an agent observes that a norm has been violated, it may decide either to sanction or to ignore the violation based on its utility. Second, an agent may invoke this component if it intends to perform an action. The agent may check if its intended action violates its norms (p- and g-norms). When making a decision about norms, an agent can have different strategies. It may be selfish and want to maximize its utility, or it may not follow the norm in the absence of norm enforcers. Another agent may be altruistic and may even bear the cost of punishing another agent that violates the norm. Some agents may be opportunistic norm followers, and they may violate a norm in certain situations. For example, on identifying the strength of the norm (e.g. littering is punished only 50% of the time), agents may opportunistically violate the norm. Thus, based on its decision making factors agents can decide whether to perform an action or refrain from performing it.
4. **Norm identification in computer-mediated interactions** - The norm identification mechanism proposed in this thesis and demonstrated through simulations can be applied to the identification of norms among human participants in computer-mediated interaction scenarios such as the use of Google Wave (Ferrate, 2010) and chat lines in MMOGs (Ludlow and Wallace, 2007). Depending upon the context of interaction, the symbols used in computer-mediated interactions (e.g. use of smiley’s to communicate human emotions in a chat system) can be used as a starting point for norm identification or violation detection. One of the problems in these scenarios is natural language
processing (Allen, 2003). For a simple domain, norm identification might be possible because of the limited search space. However, due to larger combined state spaces of agents in open conversation scenarios, it may be computationally expensive to identify such norms. We believe this is a good venue for further research.

5. **Explicit addition of power and normative power** - The notion of power in normative systems can be explored in more detail. Note that there is an implicit assumption in our work that there is some kind of social power structure (e.g. a waiter can sanction the customer). This is because the focus of our work is on social norms, where the individual members of the society are empowered to punish their peers in a social context. There is scope for investigating the notion of general powers (based on roles and authority) and normative powers (Jones and Sergot, 1996; Castelfranchi, 2002; Oren et al., 2010) in the future.

6. **Identification of conflicting norms** - Norm conflict resolution is being studied by some researchers (e.g. Kollingbaum et al. (2008); Vasconcelos et al. (2009)). We believe the work reported in this thesis can be used to identify conflicting norms. For example, the mechanisms reported in this work can be used to identify conflicting norms such as one being obliged to drink in a party\(^4\) \((O_X(drink))\) and also is expected to abstain from drinking \((F_X(drink))\) if one was to drive home after the party.

7. **Identification of complex conditional norms** - The thesis describes how simple conditional norm can be identified. Further work on identifying complex norms (e.g. norm with multiple conditions) can be undertaken.

### 10.3 Summary

This chapter summarized the contributions of this thesis. It described the limitations of the norm identification aspect of this thesis and provided pointers on how these can be overcome. It also described the future research directions in the area of norm identification.

\(^4\)Drink implies consuming alcoholic drinks in a party.
Chapter 11

Conclusion

Norms have been of interest to researchers in diverse fields because they facilitate social control which enables smoother functioning of societies. With the advent of digital societies such as Second Life, the need for establishing social order using mechanisms such as norms has grown. In the field of multi-agent systems, researchers have investigated how norms can be used to facilitate cooperation and coordination. They have designed architectures and mechanisms for the creation, spreading and the emergence of norms. However, two aspects that have not received much attention are a) a distributed leadership-based approach for norm emergence and b) norm identification in an agent society. This thesis makes contributions to these two aspects of the study of norms. The contributions to these two aspects are from two viewpoints, the bird’s eye view (emergence approach) and the viewpoint of a single agent.

The first part of the thesis is on norm emergence. The emergence approach of norms studied in this thesis makes use of a role model mechanism. The role model mechanism facilitates the creation of a hierarchy of leaders where a leader has followers who in turn are followed by other agents. The thesis also investigates how norms emerge on top of different types of network topologies. In particular, this thesis applies the particle-collision model for the construction of dynamic network topologies and also extends it to model how two different societies can be brought together.

The second part of this thesis answers the question of how an agent can recognize norms in open agent societies. An internal agent architecture for norm identification has been proposed and studied. In particular, mechanisms for the identification of two types of norms,
prohibition norms and obligation norms, have been presented. The identification algorithms make use of association rule mining, a data mining approach for identifying norms. Experimental results on the identification of prohibition and obligation norms have been discussed in the context of norms against littering a park and the tipping norm widely followed in restaurants, respectively. We have demonstrated that an agent is able to dynamically adapt the parameters of norm inference mechanism to identify norms. We have shown that an agent is able to add, remove, and modify norms in an open agent society where the norms might be changing. We have also demonstrated, using a simple scenario, how conditional norms can be identified. The architecture discussed in this thesis can be used to identify co-existing norms and also can potentially be used to identify conflicting norms. The limitations and future research directions have been discussed.

We believe that the norm emergence and norm identification mechanisms presented in this thesis can be applied to virtual agent societies, such as Second Life and massively multi-player online games. For example, emergence of norms can be facilitated through the role model mechanism in Second Life, and the norm identification mechanism can be used by an agent to detect new norms in massively multi-player online games.
References


Paolo Busetta, Ralph R


Doris M. Dehn and Susanne van Mulken. The impact of animated interface agents: a review


Jiawei Han and Micheline Kamber. *Data mining: concepts and techniques*. Morgan Kaufmann, 2006.


Michael Rymaszewski, Wagner James Au, Mark Wallace, Catherine Winters, Cory On-
drejka, Benjamin Batstone-Cunningham, and Philip Rosedale. Second Life: The Official

Fariba Sadri, Kostas Stathis, and Francesca Toni. Normative kgp agents. Computational &
Mathematical Organization Theory, 12(2-3):101–126, 2006. ISSN 1381-298X.

Norman Salazar, Juan A. Rodríguez-Aguilar, and Josep L. Arcos. Infection-based norm

Norman Salazar, Juan A. Rodríguez-Aguilar, and Josep L. Arcos. Handling uncertainty in
the emergence of social conventions. In Third IEEE International Conference on Self-

Giovanni Sartor. Why agents comply with norms and why they should. In Rosaria Conte
and Chrysanthos Dellarocas, editors, Social Order in Multiagent Systems, pages 19–44.

Christian Scheve, Daniel Moldt, Julia Fix, and Rolf Luede. My agents love to conform:
Norms and emotion in the micro-macro link. Computational and Mathematical Organi-
zation Theory, 12(2–3):81–100, 2006. ISSN 1381-298X.


Onkur Sen and Sandip Sen. Effects of social network topology and options on norm emer-
gence. In Julian Padget, Alexander Artikis, Wamberto Vasconcelos, Kostas Stathis, Vi-
viane da Silva, Eric Matson, and Axel Polleres, editors, Coordination, Organizations,
Institutions and Norms in Agent Systems V, volume 6069 of Lecture Notes in Computer

Sandip Sen and Stephane Airiau. Emergence of norms through social learning. In Proceed-
ings of the Twentieth International Joint Conference on Artificial Intelligence (IJCAI),

230


Viviane Torres Silva. From the specification to the implementation of norms: an automatic approach to generate rules from norms to govern the behavior of agents. *Autonomous Agents and Multi-Agent Systems*, 17(1):113–155, 2008. ISSN 1387-2532.


Michael Wooldridge, Nicholas R. Jennings, and David Kinny. The Gaia methodology for agent-oriented analysis and design. Autonomous Agents and Multi-Agent Systems,


Appendix A

Graphs on norm emergence

A.1 Varying Time-To-Live (TTL) when two societies are brought together (TTL = 200)

Figure A.1: Graph showing the norm convergence results when two societies of different sizes are brought together (TTL = 200)
A.2 Varying Time-To-Live (TTL) when two societies are brought together (TTL = 50)

Figure A.2: Graph showing the norm convergence results when two societies of different sizes are brought together (TTL = 50)
Appendix B

Terms used in the chapters on norm identification

Acronyms and expansions of the terms used in Chapters 6 to 9 of this thesis are given in Table B.1 based on the alphabetical ordering of the acronyms.
<table>
<thead>
<tr>
<th>Acronym</th>
<th>Expansion</th>
</tr>
</thead>
<tbody>
<tr>
<td>CONS</td>
<td>Candidate Obligation Norm Set</td>
</tr>
<tr>
<td>CPNS</td>
<td>Candidate Prohibition Norm Set</td>
</tr>
<tr>
<td>EEL</td>
<td>Event Episode List</td>
</tr>
<tr>
<td>ES</td>
<td>Event Sequences</td>
</tr>
<tr>
<td>HL</td>
<td>History Length</td>
</tr>
<tr>
<td>NEEL</td>
<td>Normal Event Episode List</td>
</tr>
<tr>
<td>NIT</td>
<td>Norm Identification Threshold</td>
</tr>
<tr>
<td>NIF</td>
<td>Norm Inference Frequency</td>
</tr>
<tr>
<td>NREES</td>
<td>Norm Related Event Episode Set</td>
</tr>
<tr>
<td>ONI</td>
<td>Obligation Norm Inference</td>
</tr>
<tr>
<td>OP</td>
<td>Occurrence Probability</td>
</tr>
<tr>
<td>SEE</td>
<td>Special Event Episode</td>
</tr>
<tr>
<td>SEEL</td>
<td>Special Event Episode List</td>
</tr>
<tr>
<td>SEL</td>
<td>Sub Episode List</td>
</tr>
<tr>
<td>SES</td>
<td>Special Event Set</td>
</tr>
<tr>
<td>tempEEList</td>
<td>temporary Event Episode List</td>
</tr>
<tr>
<td>UES</td>
<td>Unique Event Set</td>
</tr>
<tr>
<td>WS</td>
<td>Window Size</td>
</tr>
</tbody>
</table>

Table B.1: Acronyms used and the corresponding expansions