GAVEL: A Sanction-Based Regulation Mechanism for Normative Multiagent Systems

Igor Conrado Alves de Lima

THESIS SUBMITTED
TO THE
INSTITUTE OF MATHEMATICS AND STATISTICS
AT
UNIVERSITY OF SÃO PAULO
FOR THE DEGREE
OF
MASTER OF SCIENCE

Programme: Computer Science
Advisor: Professor Dr Jaime Simão Sichman
Co-advisor: Professor Dr Luis Gustavo Nardin

This research was supported by CNPq, Brazil, grant number 131120/2016-6.

São Paulo, November, 2019
GAVER: A Sanction-Based Regulation Mechanism for Normative Multiagent Systems

Esta versão da dissertação contém as correções e alterações sugeridas pela Comissão Julgadora durante a defesa da versão original do trabalho, realizada em 06/11/2019. Uma cópia da versão original está disponível no Instituto de Matemática e Estatística da Universidade de São Paulo.

Comissão Julgadora:

- Prof. Dr Jaime Simão Sichman (orientador) - IME-USP
- Prof. Dr Jomi Fred Hübner - UFSC
- Prof. Dr Juliano de Souza de Albuquerque - FD-USP
Acknowledgements

Firstly, I would like to thank my advisor, Prof. Dr Jaime Simão Sichman, for the opportunity of working with you towards this endeavour. Thank you for all the enlightening discussions, for Dr so patient and understanding, and for the many times you moved out of your way to help me both academically and personally. For all this and a lot more you have done, thank you!

Likewise, I would like to thank my co-advisor, Prof. Dr Luis Gustavo Nardin. Your guidelines were paramount for the development of this work and me as a graduate student. Thanks for having made the physical distance between us almost inexistent and being so meticulous about everything. I also feel honoured for having worked with you. Thank you!

I would also like to thank CNPq for having financially supported this research.

Thank you all my friends and colleagues with whom I had the opportunity to share great moments during this stage of my life. A special thanks to those from the Laboratory of Intelligent Techniques (LTI) and POLI USP Jiu-Jitsu Team with whom I was always learning.

Last but not least, I would like to thank my family who, despite the long distance, was always there for me. You were very important to me during the entire journey to reach this goal.
Resumo

DE LIMA, I. C. A. *GAVEL: Um Mecanismo de Regulação Baseado em Sanções para Sistemas Multiagentes Normativos*. 2019. Dissertação (Mestrado) - Instituto de Matemática e Estatística, Universidade de São Paulo, São Paulo, 2019.

O uso de uma abordagem normativa para governar sistemas multiagentes (MAS) tem sido motivada pelo interesse crescente em balancear autonomia de agentes e controle sistêmico. Em sistemas multiagentes normativos (NMAS), apesar da existência de normas especificando regras sobre como agentes devem ou não devem se comportar, agentes têm autonomia para decidir se irão agir ou não de acordo com as normas. Uma maneira apropriada de governar agentes é utilizando mecanismos de regulação baseado em sanções. Tais mecanismos permitem a autonomia dos agentes enquanto mantêm um certo nível de controle do sistema através da aplicação de sanções. No entanto, a maioria dos mecanismos de regulação encontrados na literatura não provêm suporte para associação de diferentes categorias e valores de sanções ao cumprimento e violação de normas, portanto abstenho agentes da capacidade de raciocinar e decidir sobre sanções. Uma exceção é o modelo proposto por Nardin *et al.* (2016). Baseado neste último, esta dissertação apresenta um arcabouço operacional de aplicação de sanções, denominado *GAVEL*, que habilita agentes a decidir por sanções mais apropriadas a serem aplicadas com base em fatores contextuais de decisão. As vantagens de aplicação de diferentes categorias e magnitude de sanções são ilustradas por um estudo de caso do domínio da economia, nomeadamente o jogo dos bens públicos (PGG). Os resultados experimentais mostram que possibilitar agentes decidirem entre sanções materiais e sociais leva a taxas de cooperação similares mas maiores níveis de riqueza em comparação com o uso exclusivo de sanção material.

Palavras-chave: Sistemas multiagentes normativos, regulação, sanção, engenharia de software.
Abstract

DE LIMA, I. C. A. *GAVEL: A Sanction-Based Regulation Mechanism for Normative Multiagent Systems*. 2019. Thesis (Masters Degree) - Instituto de Matemática e Estatística, Universidade de São Paulo, São Paulo, 2019.

The use of a normative approach to govern multiagent systems (MAS) has been motivated by the increasing interest in balancing between agents’ autonomy and global system control. In normative multiagent systems (NMAS), despite the existence of norms specifying rules about how agents ought or ought not to behave, agents have the autonomy to decide whether or not to act in compliance with such norms. A suitable way to govern agents is using sanction-based enforcement mechanisms. These mechanisms allow agents autonomy while maintaining a certain system control level through sanction applications. However, most enforcement mechanisms found in the literature lack support for the association of norm compliance or violation to different sanction categories and strength, thus refraining agents from sanction reasoning and decision capabilities. An exception is the model proposed by Nardin et al. (2016). Based on this latter model, this thesis presents an operational sanctioning enforcement framework, named *GAVEL*, which endows agents with the capability to decide for the most appropriate sanctions to apply, depending on their context assessed by a set of decision factors. The advantages of applying different sanctions categories and strength are illustrated by a case study from the domain of economics, namely the public goods game (PGG). The experimental results show that allowing agents to decide between material and social sanctions leads to similar cooperation rates but greater wealth levels in comparison to solely using material sanction.

Keywords: Normative multiagent systems, regulation, sanction, software engineering.
Contents

List of Abbreviations ix
List of Figures xi
List of Tables xiii

1 Introduction 1
   1.1 Motivation ...................................................... 2
   1.2 Objectives ..................................................... 2
   1.3 Contributions .................................................. 2
   1.4 Methodology .................................................... 3
   1.5 Thesis Structure ............................................... 3

2 Foundations and State-of-the-art 5
   2.1 Normative Multiagent System .................................. 5
   2.2 Sanctioning Enforcement ........................................ 6
      2.2.1 Trust and Reputation ...................................... 6
      2.2.2 Norm enforcement .......................................... 6
   2.3 Sanctioning Enforcement Model ................................ 7
      2.3.1 Sanction Typology .......................................... 7
      2.3.2 Sanctioning Process ....................................... 9
   2.4 Prospect Theory ................................................ 10
      2.4.1 Cumulative Prospect Theory ............................... 11
      2.4.2 Reference-Dependent Preferences ....................... 12
   2.5 MAS Programming ............................................... 13
      2.5.1 Programming Agents ...................................... 14
      2.5.2 Programming Environments ................................ 14
      2.5.3 Programming Organisations ................................ 15
      2.5.4 The JaCaMo Framework ................................... 15
   2.6 Discussion ...................................................... 16

3 GAVEL Framework 19
   3.1 Formal Definition ............................................... 19
      3.1.1 NMAS ....................................................... 19
      3.1.2 Norms, Sanctions and Links ............................. 20
      3.1.3 Repositories .............................................. 21
## CONTENTS

3.1.4 Capabilities ............................................. 22  
3.1.5 Sanctioning Process ......................................... 23  
3.2 Implementation ........................................... 24  
3.3 Discussion ................................................ 26  

4 Case Study ............................................... 29  
4.1 Public Goods Game (PGG) ..................................... 29  
4.1.1 Overview ................................................. 29  
4.1.2 Contribution Strategies .................................... 30  
4.1.3 Sanctioning Strategies ...................................... 32  
4.2 Implementation ............................................. 32  
4.3 Experiments ............................................... 34  
4.3.1 Description ............................................... 35  
4.3.2 Cooperation rate results ................................... 35  
4.3.3 Sanction application results .............................. 36  
4.3.4 Surviving agent results ................................... 38  
4.3.5 Wealth results ............................................ 39  
4.3.6 Statistical Analysis ........................................ 40  
4.4 Discussion ................................................ 43  

5 Conclusions and Future Work .......................... 45  
A Regulative specification schema file ..................... 47  
B Capabilities specification schema file ................. 51  

References ............................................... 53
List of Abbreviations

| Abbreviation | Full Form                                    |
|--------------|----------------------------------------------|
| MAS          | Multiagent system(s)                        |
| NMAS         | Normative multiagent system(s)              |
| PGG          | Public goods game                            |
List of Figures

2.1 Dimensions of the sanction typology proposed by Nardin et al. (2016) ........................................ 7
2.2 Sanctioning enforcement model (Nardin et al., 2016) ................................................................. 9
2.3 Cumulative prospect theory’s value function .................................................................................. 12
2.4 Non-linear weighting function found by Tversky and Kahneman (1992) as part of the cumulative prospect theory ................................................................. 13
2.5 Overview of a JaCaMo multiagent system, highlighting its three dimensions (Boissier et al., 2013b) ................................................................. 16
3.1 Sanctioning process ......................................................................................................................... 24
3.2 GAVELE’s architecture .................................................................................................................... 25
4.1 Round with all players complying with the norm .......................................................................... 33
4.2 Round with one player violating the norm .................................................................................... 33
4.3 Cooperation rates within the punishment sanctioning strategy experimental group .................. 36
4.4 Cooperation rates within the mixed sanctioning strategy experimental group ............................ 36
4.5 Number of material sanctions applied over the rounds in the punishment experimental group .................................................................................................................. 37
4.6 Number of material sanctions applied over the rounds in the mixed experimental group ........ 37
4.7 Number of gossips made over the rounds in the mixed experimental group ............................... 37
4.8 Average number of sanctions applied in each experiment of the punishment group ................ 38
4.9 Average number of sanctions applied in each experiment of the mixed group .......................... 38
4.10 Average number of survivor agents in each experiment of the punishment group ..................... 39
4.11 Average number of survivor agents in each experiment of the mixed group ............................. 39
4.12 Average population wealth in each experiment of the punishment group ..................................... 39
4.13 Average population wealth in each experiment of the mixed group ........................................... 40
4.14 Box plots representing the distribution of cooperation rates for each experiment ...................... 41
4.15 Bar plots illustrating the average population wealth in the end of simulations of each experiment .................................................................................................................. 43
# List of Tables

2.1 Summary of requirements fulfilled by the existing sanctioning enforcement frameworks.  
    Adapted from Nardin (2015). .................................................. 17

4.1 Simulation parameters. ......................................................... 35
4.2 Combinations of parameters used in the PGG experiments. ............... 35
4.3 Mann-Whitney U test of cooperation rates. .................................. 41
4.4 One-sided Mann-Whitney U test of number of material sanctions. ......... 42
4.5 Mann-Whitney U test of final population wealth. ............................ 43
Chapter 1

Introduction

A multiagent system (MAS) is characterised by a set of heterogeneous agents that interact (e.g., cooperate, compete, and negotiate) in order to perform their tasks and achieve their goals (Wooldridge, 2009). An important characteristic of agents in MAS is the capability to operate autonomously. If not properly governed, however, such autonomy may result in degraded emerging properties of the system (Johnson et al., 2012).

For the last two decades, the normative approach has attracted attention of the scientific community as a means to tackle the issue of MAS governance. Such interest is due to the fact that norms have been recognised as playing a key role in regulating humans’ behaviours and maintaining the social order in human society (Andrighetto et al., 2013; Boella et al., 2006, 2008; Castelfranchi, 1995, 2000; Conte et al., 1998; Hollander and Wu, 2011; Verhagen, 2000). In agreement with Boella et al. (2006), we refer to norms as “a principle of right action binding upon the members of a group and serving to guide, control, or regulate proper and acceptable behaviour”, and to normative as a qualifier of something “conforming to or based on norms”. Normative multiagent systems (NMAS) are then the integration of normative concepts into MAS. This approach has been motivated by the increasing interest in balancing between agents’ autonomy and control in MAS (Verhagen, 2000).

In NMAS, norm enforcement enables reaction to norms violation (e.g., punishment) or compliance (e.g., reward) henceforth identified as sanction. The degree to which a norm is enforced plays a crucial role in NMAS dynamics and it conveys a great deal of norm-relevant information that affects other normative processes.

There are two traditional approaches to norm enforcement in NMAS (Nardin et al., 2016):

- the regimentation approach (also known as off-line norm enforcement) which assumes that agents can be controlled and non-compliant actions can be prevented;
- the regulation approach (also known as on-line norm enforcement) in which violations may occur, yet whenever a violation is detected, sanctions may be applied to the violator.

Let us consider a scenario in which there is a norm prohibiting an agent $A$ from sending messages during a certain period. Using regimentation, if $A$ violates the norms and tries to send a message at a forbidden time, some entity (e.g., communication middleware) would intercept the message, preventing the violation effects from taking place in the environment. Conversely, in the regulation approach, when $A$ violates the norm, the message is effectively sent in the environment, but $A$ becomes liable to be sanctioned by other agents for such violation.

Both norm enforcement approaches can be arranged in a centralised or distributed mode. The regimentation approach operates mostly in a centralised mode through normative institution frameworks such as Electronic Institutions (Esteva et al., 2000; Kafalý et al., 2016; Noriega, 1997) and Organisation Models (Dignum, 2004; Gâteau et al., 2005). These frameworks provide a reference normative system to which agents have to abide and infrastructure entities enforce the compliance of agents’ actions and interactions with the norms (Fornara and Colombetti, 2008; Grossi et al., 2006). The regulation approach, conversely, operates in a distributed mode; from the agents’ point of view, architectures such as BOID (Broersen et al., 2001), NoA (Kollingbaum and Norman, 2003), ...
and EMIL-A (Andrighetto and Villatoro, 2011) allow them to reason and decide whether or not to follow norms. This work focuses particularly on the regulative approach for norm enforcement.

1.1 Motivation

Norms by themselves do not guarantee that agents will comply with them. Fundamentally, agents have the autonomy to decide whether or not to comply with such norms. The governance process of a NMAS thus needs a framework to motivate norm compliance. For this reason, this work focuses on the use of a regulative approach sustained by sanctions as a means to steer compliance.

Sanctions are the basis for the regulative approach in norm enforcement. In their studies, Pasquier et al. (2005) identified the importance and need for having different sanction types and endowing agents with sanction reasoning and decision capabilities. In contrast, Nardin (2015, p. 56-57) showed that the available norm enforcement frameworks lack full support to four main requirements to render these features possible:

R1 Support for multiple sanctions categories (e.g., legal sanctions, ostracism, reputation spreading);
R2 Potential association of multiple sanctions with a norm violation or compliance (e.g., provide a set of sanction options instead of pre-establishing a fixed set of sanctions to a norm violation or compliance);
R3 Reasoning about the most adequate sanction to apply depending on different factors (e.g., one might consider the sanctionee’s history to determine an appropriate sanction to apply, if any);
R4 Adaption of the sanction content depending on context (e.g., a norm violation of high magnitude might incur a more severe negative sanction).

Existing sanctioning frameworks found in the literature suffer from drawbacks that render them unsuitable for supporting these four requirements. Therefore, there is a need for a norm enforcement framework able to build better governance in NMAS.

1.2 Objectives

The main objective of this work is to create a sanctioning enforcement framework, named GAVEL, which addresses the requirements identified by Nardin (2015) in order to endow agents with sanction reasoning and decision capabilities. More specifically, this work intends to:

a) Refine and operationalise the conceptual sanctioning process model proposed by Nardin et al. (2016);
b) Demonstrate how GAVEL may be used in a social dilemma when sanctions with different categories and strength may be applied in response to a norm compliance or violation.

1.3 Contributions

This work produces the following three major contributions:

C1 Operational sanctioning enforcement model for MAS. This model formalises the main components, capabilities, and data repositories that enable agents to detect and evaluate normative behaviours, as well as specify, apply, and monitor sanctions depending on their context and goals;

C2 Sanctioning enforcement framework (GAVEL). Based on the resulting model from the previously mentioned contribution (C1), this framework is essentially independent from MAS platforms, hence seamless integration should be possible;
C3 Integration of GAVEL with a widely used MAS platform (JaCaMo). This should allow faster adoption of GAVEL by the MAS community and demonstrate how an integration with MAS platforms may be realised.

1.4 Methodology

The methods which were used to conduct this work are the following:

1. Operationalisation and refinement of the conceptual sanctioning enforcement model described by Nardin et al. (2016), introducing temporal aspects to norms (i.e., maintenance and deadline conditions), adding enablement status to associations between norms and sanctions, considering agents’ beliefs in decision making, and decoupling sanction decision, application, and outcome phases;

2. Creation and integration of a sanctioning enforcement framework (GAVEL) based on the model proposed with a MAS platform to ensure the model is indeed practical;

3. Validation of the GAVEL framework by using it to implement mixed sanction strategies in a version of the public goods game (PGG), a famous game used in experimental economics (Ledyard, 1995) wherein rational choice determines that it is better for each player not to cooperate, even though they would be better-off cooperating, thus characterising a social dilemma.

1.5 Thesis Structure

The remainder of this work presents the following structure:

- Chapter 2 discusses all the foundations for this study including a discussion about related works;
- Chapter 3 details the sanctioning enforcement framework GAVEL;
- Chapter 4 describes the case study and experimental results;
- Chapter 5 presents conclusions and future work.
Chapter 2

Foundations and State-of-the-art

This chapter presents fundamental concepts that underlie the proposal of this study. Firstly, Section 2.1 starts by describing normative multiagent systems (NMAS). Secondly, Section 2.2 presents sanctioning enforcement in NMAS by reviewing existing frameworks from the literature. Thirdly, Section 2.3 discusses the process model used as basis for this study. Then, Section 2.4 briefly introduces prospect theory, a theory in cognitive psychology that describes the way people choose between probabilistic alternatives which involve risk. Next, Section 2.5 provides a broad overview on multiagent system (MAS) programming and particularly describes JaCaMo, a widely used multiagent-oriented programming platform on which GAVEL has also been implemented. Finally, a summary of how these concepts are interconnected in this work is provided in Section 2.6.

2.1 Normative Multiagent System

A multiagent system (MAS) is a system composed of a set of autonomous and heterogeneous agents situated in a shared environment where they interact and, eventually, build or are coordinated by organisations (Ferber, 1999). MAS may be classified as closed or open systems. Closed MAS are those in which all agents know each other and interact via structured and predictable protocols following specific patterns. These systems are usually designed with a specific purpose in mind. Conversely, open MAS are general purpose systems in which (1) agents’ behaviours and interactions cannot be known in advance, (2) their internal architecture as well as beliefs and goals are not shared, and (3) they can join and leave the system at any time (Artikis and Pitt, 2008; Hewitt, 1991). Open MAS properties hinder assurance that all agents will behave as expected in a way that the system exhibits desirable global properties (e.g., stability and efficacy). Thus, the use of certain mechanisms to steer the system in a preferred direction becomes very important, yet maintaining a certain level of agents’ autonomy (Pasquier et al., 2006). One possible strategy to achieve this goal is governing agents’ behaviours through normative systems (i.e., normative constraints), as in human societies.

In computer science, normative systems are redefined as those in which “norms play a role and which need normative concepts in order to be described or specified” (Meyer and Wieringa, 1993). Drawing upon Wright (1963), and a long tradition of deontic philosophy and logic-based theory of action, normative systems define the global desired properties of the system by means of norms that specify obligations, prohibitions, and permissions (Conte and Castelfranchi, 2006). Wright (1963) defined norms based on how they influence and relate to actions, distinguishing them between rules, prescriptions, customs, directives, moral principles, and ideals.

Normative multiagent systems (NMAS) revolve around the idea that, as in human societies, individual and collective agent behaviours are affected (i.e., governed) by norms. They are a combination of normative systems and MAS, aiming to govern MAS and establishing the balance between agents’ interests and the desired global system’s properties (Boella and van der Torre, 2003; Castelfranchi and Falcone, 1998; Shoham and Tennenholtz, 1992; Verhagen, 2000).

The next section presents sanction enforcement frameworks and how they have been used in
2.2 Sanctioning Enforcement

There are two complementary sanctioning approaches to achieve social control and order in MAS. These approaches are known as trust and reputation and norm enforcement, each of which are discussed next.

2.2.1 Trust and Reputation

Trust and reputation reflect the idea of indirect sanctioning in which agents, instead of acting directly against others (e.g., imposition of fines), use information about the past behaviour of others to evaluate how they might perform in the future and decide whether and how to interact with them. Such historical information that agents communicate amongst themselves about a certain target is regarded as reputation transmission. In this sense, reputation arises as a key component of trust, emerging implicitly in a society as a social control mechanism (Conte and Paolucci, 2002; Pinyol and Sabater-Mir, 2013). A positive performance history thereby would ordinarily lead to higher trust that the agent will perform well in the future, whereas a negative history results in the opposite. Thus, a sanction would arise indirectly via future actions.

Due to the importance of trust and reputation for MAS (Castelfranchi and Falcone, 1998), several models have been proposed in the literature in the last two decades. The following non-exhaustive list is representative of trust and reputation models available in the MAS literature: Zacharia and Maes (2000), Mui et al. (2003), ReGreT (Sabater and Sierra, 2001), Repage (Conte and Paolucci, 2002; Sabater et al., 2006), FIRE (Dong-Huyhnia et al., 2004), Wang and Singh (2010), L.I.A.R. (Vercouter and Muller, 2010), and BDI + Repage (Pinyol et al., 2012). Further information about computational trust and reputation models can be found on Pinyol and Sabater-Mir (2013) and Hendrikx et al. (2015).

2.2.2 Norm enforcement

The process in which agents monitor and encourage others to comply with the norms is referred to as norm enforcement. The degree to which a norm is enforced plays a crucial role in its dynamics as it conveys a great deal of norm-relevant information that affects other normative processes, such as norm recognition, adoption and compliance (Conte et al., 2013; Nardin, 2015). Thus, norm enforcement is a reinforcement framework that guarantees the stability and robustness of the norm life cycle.

In the regulative approach for norm enforcement, the two traditional approaches to the enforcement of norms are:

- **Institutional (centralised)**, which assumes a central high-level authority that observes, controls or enforces agents’ actions and interactions, and sanctions them in case of normative behaviours;

- **Social (decentralised)**, which endows agents with mechanisms to monitor and evaluate others’ behaviours, as well as apply sanctions whenever appropriate.

Since agents are autonomous and pursue their own goals, norm internalisation is one possible explanation of why agents comply with norms even in situations they would be better-off violating them. A norm is internalised whenever its maintenance has become independent of external support events, such as reinforcement through sanctions (Aronfreed, 1969).

Norm enforcement frameworks play an important function in norm internalisation and in directing the system towards an expected path via the process of enforcement. Cardoso and Oliveira (2009) proposed a centralised norm enforcement framework for contractual commitments. Their solution pre-define sanctions that are applied by enforcer agents regardless of any individual or contextual information. Centeno et al. (2011) extended this approach to adapt sanctions based on contextual information. Modgil et al. (2009, 2015) proposed a general distributed architecture for norm-governed
2.3 SANCTIONING ENFORCEMENT MODEL

The sanctioning enforcement model proposed by Nardin et al. (2016) is a conceptual model which aims to enable agents to reason about and adapt their sanctions. This model is based on the social approach in which the agents themselves are responsible for performing an adaptive and auto-organised peer control. For such purpose, agents are endowed with mechanisms to monitor their peers, assess their behaviours, and apply sanctions whenever necessary.

The model adopts a sanction typology, which lays the foundation for a comprehensive notion of sanctions, as described next.

2.3.1 Sanction Typology

Based on an extensive and interdisciplinary study ranging from sociology to socio-technical systems, Nardin et al. (2016) propose a comprehensive sanction typology composed of six dimensions, as shown in Figure 2.1. These dimensions are purpose, issuer, locus, mode, polarity, and discernability.

These dimensions are described in terms of source, target, sender, and receiver agents. Source and target relate to the sanction content. Source refers to the agent who generates the sanction (i.e., a violation reporter), whereas target indicates the agent to whom the sanction is directed (i.e., the accused). In contrast, sender and receiver relate to the agents who apply and receive the sanction, respectively.

In order to illustrate how these terms differ, suppose a situation involving the agents Alice, Bob, and Carol. Suppose that Alice sanctions Carol by informing Bob that Carol is not trustworthy. In
this case, Alice is both the source and the sender, as it generates and applies the sanction; Bob is the receiver, as it receives and process the sanction; and Carol is the target, as it is the one to whom the sanction applies.

### Purpose

The purpose specifies the effect the sanction is expected to have on the social environment. The purpose may fall into one of five categories, organised in two different aspects.

1. The *influence* aspect deals with incentives, negative or positive, and covers two purposes for a norm violation or compliance: *punishment* seeks to penalise targets in order to prevent non-compliant behaviour; *reward* seeks to promote and motivate targets towards compliant behaviour.

2. The *performance* aspect deals with capabilities and covers three purposes: *incapacitation* seeks to restrict the target’s action making the norm violation impossible for a bounded period of time\(^1\); *guidance* seeks to change the target’s conduct by instructing how to comply with the norm; *enablement* seeks to provide opportunity, and possibly the mean, for the target to comply with the norm.

### Issuer

The issuer specifies whether the sanction’s enforcer or issuer is a formally acknowledged authority. The issuer dimension may fall into one of two categories: *formal* sanctions are officially enforced by generally recognised authorities, such as the State, regulatory agencies (e.g., Federal Reserve Board), or traders (e.g., AliExpress); *informal* sanctions are unofficially enforced by members of the society such as ridicule, ostracism, praise, and others.

### Locus

The locus specifies the recipient of a sanction determining if it is *self-directed* (i.e., the sender and receiver are the same) or *other-directed* (i.e., the sender and receiver differ) regarding the individual who applies the sanction, the sender. A self-directed sanction is aimed at the sender itself (e.g., vicarious shame) whereas an other-directed sanction is aimed at an individual different from the sender (e.g., fine imposition).

### Mode

The mode specifies how a sanction should affect its target. A *direct* sanction should affect the target directly and immediately (e.g., self-blame, fine), whereas an *indirect* sanction affects its target indirectly, potentially affecting others’ future actions which might, in turn, affect the target (e.g., gossip).

### Polarity

The polarity of a sanction specifies its content in terms of if it is *positive* or *negative*. A positive sanction indicates a reward (e.g., praise) and a negative sanction indicates a penalty (e.g., fine).

### Discernability

The discernability specifies how perceptible a sanction is to its target. A *noticeable* sanction is one that forces a target to notice it (e.g., Alice thanking Bob for helping her achieving a contractual goal), whereas an *unnoticeable* sanction is not easily noticeable by its target (e.g., Alice badmouthing Carol).

---

\(^1\)The incapacitation dimension differs from regimentation as, in the latter, it is *always* impossible to violate a norm, rather than only for a period of time.
2.3.2 Sanctioning Process

Figure 2.2 depicts the sanctioning process conceptual model. It is composed of five capabilities (Detector, Evaluator, Executor, Controller, and Legislator) using two resources (De Jure and De Facto data repositories). Capabilities are active entities, whereas resources are passive entities. These capabilities and resources may be realised in multiple ways, including in a fully centralised or a fully decentralised manner.

The repositories are used to store information about norms and sanctions. The De Jure repository stores norms and sanctions, as well as links between them, which are also known by the agents. One norm may be related to different sanctions, whereas one sanction may be triggered by different norms. The De Facto repository stores information about the applied sanctions and other relevant information used to assess their efficacy.

The five capabilities of the sanctioning enforcement process are:

- Detector – perceives the environment and detects any norm violation or compliance, and sanctions applied by other agents;
- Evaluator – obtains information from De Jure and De Facto to determine whether and which sanctions to apply;
- Executor – applies a sanction;
- Controller – monitors the outcomes of applied sanctions to evaluate their efficacy and records information about sanctions applied by other agents;
- Legislator – updates De Jure based on an assessment of De Jure and De Facto.

Consider a public transport system scenario\(^2\) to illustrate how these capabilities and resources work. The Department of Transportation of a legal state has created the official Transport Operations

---

\(^2\)This scenario is inspired by the public transport system from the state of Queensland, Australia – https://translink.com.au/.
Regulation containing the rights and obligations of public transport passengers. One of the obligations states that free-riders (fare evaders) must pay a fine of a certain amount. To avoid such fine, every passenger must have an electronic transport card which she must touch onto a card reader at the beginning and end of every journey. The due fare is automatically calculated at the end of the journey when the card is touched onto the reader.

Trained and authorised officers are responsible for enforcing that passengers of public transports follow the regulation. Infringement notices, including fines, may be issued for public transport offences. These officers normally act by entering a (discretionary) public vehicle and approaching en-route passengers to check if their transport cards are valid for that journey. Sometimes, regulation compliant passengers also report others’ misconducts via phone call or direct contact with officers, thus helping to enforce the regulation. Every occurrence is registered in a database accessible to the Department of Transportation, which uses the data to evaluate overall performance and, possibly, update the regulation in order to protect revenue.

This scenario easily relates to the sanctioning process model described above. The Department of Transportation has the capabilities of a Legislator as it is responsible for administering the Transport Operations Regulation, that is De Jure. As passengers and officers are able to recognise misconducts (i.e., normative behaviours), decide for and apply sanctions accordingly (i.e., report and fine, respectively), they all have the capabilities of Detector, Evaluator, and Executor. A passenger may not only remember every time she did or did not make a report but also evaluate the efficacy of her acts (sanctions) under given circumstances, thus showing capabilities of a Controller. Different instances of De Facto may as well serve the purpose of storing sanction applications and outcomes, thus representing each passenger’s official database of sanctions.

It is important to notice the flexibility this model provides. The agents’ capabilities, for example, are not limited to those aforementioned. In fact, agents may possess any number of capabilities, including others not defined in the model, and use different control strategies for each component depicted in Figure 2.2. For example, one could use a prospect theory – a model that describes how agents decide between probabilistic alternatives that involve risk – to support sanction decision, thus enriching the Evaluator capability. However, this work explores the use of prospect theory for other decision making purposes in the MAS experiments reported in Chapter 4. Next section provides more details about such theory.

### 2.4 Prospect Theory

Prospect theory\textsuperscript{3} is a model in cognitive psychology that shows how people decide between alternatives that involve risk and uncertainty. Its original model emerged in 1979 by Kahneman and Tversky (1979) demonstrating that people weight gains and losses differently. In fact, the key elements of this theory are 1) a value function that is concave for gains but convex and steeper for losses, and 2) a non-linear transformation of the probability scale which overweights small probabilities and underweights moderate and high probabilities. Although the original paper has all the theory’s essential insights and has been considered a “seminal paper in behavioural economics” (Shafir and LeBoeuf, 2002), the reported model has some limitations: the model only applies to decisions between at most two non-zero outcomes, and it predicts that people tend to choose dominated decisions (Barberis, 2013). Later, in 1992, Tversky and Kahneman (1992) published a new version of prospect theory that extends the original work in several respects and resolves the problems mentioned before. This new version is called cumulative prospect theory and is the one typically used in economic analysis. This is the version which is briefly reviewed in the following section.

\textsuperscript{3}Due to its significance for economics, prospect theory rendered the 2002 Nobel Memorial Prize in Economics award to one of its creators, Daniel Kahneman.
2.4.1 Cumulative Prospect Theory

Consider a prospect (also known as gamble) as defined in Equation 2.1, where the notation should be read as “gain $x_i$ happens with probability $p_i$, $x_{i+1}$ with probability $p_{i+1}$, and so on”. The outcomes are arranged in increasing order, so that $x_i < x_j$ for $i < j$, and $x_0 = 0$. Positive values represent gains, whereas negative ones losses. In that case, a prospect expressed as $(−$100, $\frac{1}{2}$; $+$200, $\frac{1}{2}$) means a 50:50 bet to lose $100 or gain $200.

$$f = (x_{-m}, p_{-m}; x_{-m+1}, p_{-m+1}; \ldots; x_0, p_0; \ldots; x_n, p_n)$$ (2.1)

In the classical expected utility theory (Morgenstern and Von Neumann, 1953), the utility of an uncertain prospect is the sum of the utilities of the outcomes, each weighted by its probability. Formally, each individual evaluates the above prospect as Equation 2.2, where $W$ is its own current wealth and $U(\cdot)$ is an increasing and concave utility function.

$$V'(f) = \sum_{i=-m}^{n} p_i U(W + x_i)$$ (2.2)

However, there is empirical evidence (Tversky and Kahneman, 1992) suggesting two major modifications of this theory: 1) people derive utility from gains and losses, measured relative to some reference point, rather than from absolute levels of wealth; and 2) the value of each outcome should be multiplied by a decision weight, not by and additive probability.

Under the cumulative prospect theory the prospect is evaluated as Equation 2.3, where $v(\cdot)$ is an increasing value function and $\pi_i$ are decision weights. The value function, $v(\cdot)$, is concave above the reference point($v''(x) \leq 0, x \geq 0$) but convex below it ($v''(x) \geq 0, x \leq 0$), and steeper for losses than gain ($v'(x) < v'(-x), x > 0$). This formulation illustrates the four elements of prospect theory: 1) reference dependence, 2) loss aversion, 3) diminishing sensitivity, and 4) probability weighting.

$$V''(f) = \sum_{i=-m}^{n} \pi_i v(x_i)$$ (2.3)

First, in prospect theory, people derive utility from gains and losses, measured relative to some reference point, rather than from absolute levels of wealth – the argument of $v(\cdot)$ is $x_i$, not $W + x_i$. Tversky and Kahneman base this assumption, known as “reference dependence” on experimental evidence that the human perceptual system works in a similar way (Kahneman and Tversky, 1979; Tversky and Kahneman, 1992).

Second, the value function $v(\cdot)$ captures “loss aversion”, the idea that people are much more sensitive to losses than to gains of the same magnitude. A typical value function showing this behaviour may be seen in Figure 2.3, which shows a gain or loss $x$ in the horizontal axis and the value $v(x)$ assigned to it in the vertical axis. Notice that the value of a $100 gain, $v(100)$, has a smaller magnitude than the one of a $100 loss, v(-100)$. This assumption reflects why people tend to turn down prospects like $(-$50, $\frac{1}{2}$; $+$55, $\frac{1}{2})$ – $+$50 loss or $+$55 gain with 50% probability each. Similarly, people would rather choose $(-$600, $\frac{1}{2}$; $+$0, $\frac{1}{2})$ – $+$600 loss with 25% probability and $+$0 gain with 75% – over $(-$400, $\frac{1}{4}$; $-$$200, $\frac{1}{4}$; $+$0, $\frac{1}{2})$ – $+$400 loss with 25% probability, $+$200 loss with 25% probability, and $+$0 gain with 50% probability. It is hard to understand this effect under the expected utility theory. Dollar amounts are so small compared to wealth levels that the prospect is evaluated in an essentially risk-neutral manner: if the value is positive it is therefore attractive (Rabin, 2013). However, for a loss-averse individual, these prospects are unappealing. For example, the pain of losing $-50 outweighs the pleasure of gaining $55.

Third, as shown in Figure 2.3, the value function is concave in the region of gains but convex in the region of losses. This element of prospect theory is known as diminishing sensitivity because it implies that, while replacing a $100 gain (or loss) with a $200 gain (or loss) has a significant utility impact, replacing a $1,000 gain (or loss) with a $1,100 gain (or loss) has a smaller impact. The fact that people tend to be risk averse for moderate probability gains is captured by the concavity over
Foundations and State-of-the-Art

2.4

Figure 2.3: The graph plots the value function found empirically by Tversky and Kahneman (1992) as part of the cumulative prospect theory. More specifically, the function is defined as $v(x) = x^\alpha$ for $x \geq 0$ and $v(x) = -\lambda(-x)^\alpha$ for $x < 0$, where $x$ is a gain or loss. From a series of experiments with humans, the authors estimate $\lambda = 2.25$ and $\alpha = 0.88$ for the loss aversion and diminishing sensitivity coefficients, respectively. Other combinations parameters should, essentially, produce similar curves which keep the idea of loss aversion. In order to make loss aversion and diminishing sensitivity easier to visualise, the plot uses $\lambda = 2.5$ and $\alpha = 0.5$. Source: Barberis (2013).

Gains: they normally prefer a certain $50 gain rather than a $100 gain with a 50% chance. However, people also tend to be risk seeking over losses: they prefer a 50% chance of losing $50 to rather than losing $100 for sure. This motivates the convexity over losses.

The fourth and final component of prospect theory is probability weighting. In prospect theory, people do not weight outcomes by their objective probabilities $p_i$, but rather by transformed probabilities or decision weights $\pi_i$. The decision weights are computed with the help of a weighting function $w(\cdot)$ whose argument is an objective probability. The solid line in Figure 2.4 shows the weighting function proposed by Tversky and Kahneman (1992). In comparison to the dotted line corresponding to the expected utility benchmark, the weighting function overweights low probabilities and underweights high probabilities.

Other studies subsequent to Tversky and Kahneman’s (1992) have tried to use new experimental data to estimate the value function $v(\cdot)$ and decision function $w(\cdot)$ (Abdellaoui, 2000; Bruhin et al., 2010; Gonzalez and Wu, 1999). These studies confirm the original properties of these functions: the loss aversion and diminishing sensitivity features of the value function, and the inverse S-shape of the weighting function. They provide especially strong support for probability weighting.

Reference-Dependent Preferences

The central idea in prospect theory is that people derive utility from gains and losses relative to a reference point. However, it is often unclear how to define precisely what a gain or loss is, since Tversky and Kahneman (1992) offered relatively little guidance on how the reference point is determined. Some significant attempts to clarify how people think about gains and losses are the works of Kőszegi and Rabin (2006, 2007, 2009). They propose a framework for applying prospect theory in economics that is both disciplined and portable across different contexts. The most important element from their framework is the idea that people use expectations as a reference point to compute gains and losses. In this case, expectation may be defined as a belief held in the recent past about outcomes. Particularly, they propose that people derive utility from the difference between consumption and expected consumption, where the utility function exhibits loss aversion.
and diminishing sensitivity. They also assume that expectations are rational, in that they match the distribution of outcomes that people will face if they follow the plan of action that is optimal, given their expectations.

A person’s utility depends not only on her $K$-dimensional consumption bundle $c$, but also on a reference bundle $r$. For example, suppose a 2-dimensional situation of choice where a person values wealth and having shoes, $c = \{c_1, c_2\}$, where $c_1 \in \{0, 1\}$ reflects whether she has shoes and $c_2 \in \mathbb{R}$ is her dollar wealth. Conversely, $r = \{r_1, r_2\}$ is her reference expectation after making a decision under risk, where $r_1 \in \{0, 1\}$ reflects whether she has shoes and $r_2 \in \mathbb{R}$ is her dollar wealth. She has an intrinsic “consumption utility” $m(c)$ that corresponds to the outcome-based utility classically studied in economics. Overall utility is given by $u(c|r) \equiv m(c) + n(c|r)$, where $n(c|r)$ is “gain-loss utility”. Both consumption utility and gain-loss utility are separable across dimensions, so that $m(c) \equiv \sum_k m_k(c_k)$ and $n(c|r) \equiv \sum_k n_k(c_k|r_k)$, where $k$ is the dimension. Because the sensation of gain or loss due to a departure from the reference point seems closely related to the consumption value attached to the goods in question, Kőszegi and Rabin (2006) assume that $n_k(c_k|r_k) \equiv \mu(m_k(c_k) - m_k(r_k))$, where $\mu(\cdot)$ satisfies the properties of Tversky and Kahneman’s (1992) value function.

For Kőszegi and Rabin (2009), the sensation of gain or avoided loss from having more money significantly affects people’s utility. Similarly, the absolute pleasure of consumption purchased with money also affects people’s utility. Thus, in evaluating an outcome, the decision-maker assesses gain-loss utility in each dimension separately. In combination with loss aversion, this separability is at the crux of many implications of reference-dependent utility, including the endowment effect.4

This model of Kőszegi and Rabin (2006) based on Tversky and Kahneman’s (1992) prospect theory is used later in Chapter 4 to model how certain agents decide to cooperate in a public goods game designed as a MAS.

2.5 MAS Programming

A MAS provides a repertoire of high-level concepts and abstractions to model and develop complex, distributed, and open systems (Boissier et al., 2013a; Dastani, 2014). At the highest level

4Endowment effect is the finding that people are more likely to retain an object they own than acquire that same object when they do not own it (Morewedge and Giblin, 2015).
of abstraction, a MAS is conceived as consisting of communicating agents, an environment with which agents interact, and an organisation that regulates the agents’ behaviours. Each of these entities are defined in terms of other concepts and abstractions such as beliefs, goals, plans, actions, events, roles, interaction, organisational rules and structures, communication, norms, and regulation policies.

The main aim of the multiagent programming research field is to propose programming languages and development frameworks that facilitate direct and effective implementations of MAS (Dastani, 2014). Both provide computational constructs and tools to implement specific concepts and abstractions that are proposed by some multiagent architectures.

2.5 Programming Agents

An essential characteristic of agents is autonomy. An agent is considered autonomous if it has a decision making component that determines its decisions based on informational (e.g., beliefs, distributions, knowledge), motivational (e.g., desires, utilities, preferences), and deliberational (e.g., intention, plans, commitments) attitudes. The decision-making component should allow a programmer to implement different decision issues, such as decision strategies, resolving decision conflicts, and rationality or realism of decisions (Dastani, 2014). There are various abstract decision models that may be used to implement an agent’s decision component. For example, one may implement an agent’s decision component based on decision models such as POMDP (Partially Observable Markov Decision Process model) (Tasaki et al., 2010), BDI (Belief-Desire-Intention) (Cohen and Levesque, 1990; Meneguzzi et al., 2015; Rao and Georgeff, 1991), or a combination of both (Nair and Tambe, 2005).

One of the first agent-oriented programming languages to arise was Agent-0 (Shoham, 1993). Its underlying idea was to implement agents in terms of mental components such as beliefs, commitments, capabilities, and actions. An agent program in Agent-0 consists of an initial belief base, a set of capabilities, and a set of commitment rules along with private actions that agents may perform. The execution of an agent is a continuous cyclic process wherein, at each cycle, received messages are processed, commitments are generated, and actions are performed. However, the state of an Agent-0 agent lacks motivational attitudes such as utility, desires, goals, or preferences such that an agent’s decisions are based only on events and messages rather than on its motivational attitude.

Since Agent-0 was introduced, a plethora of other agent-oriented programming languages with a larger repertoire of concepts and abstractions emerged. Some of these languages have an imperative programming style, some are declarative, and yet others combine both styles. Besides Agent-0, other platforms that use agent-oriented programming languages include 3APL (Hindriks et al., 1999), 2APL (Dastani, 2008), Jason (Bordini et al., 2007), JACK (Busetta et al., 1999; Winikoff, 2005), and GOAL (Hindriks, 2009).

The notions of belief, desire and intention (BDI) are key components in these languages. They denote what the agent believes, what it would like to achieve, and what it is currently working towards achieving. The origins of the BDI model lie in the theory of human practical reasoning developed by the philosopher Bratman (1987), which focuses particularly on the role of intentions in practical reasoning.

2.5.2 Programming Environments

In MAS applications, individual software agents interact with an environment consisting of shared resources or services (Boissier et al., 2013a; Dastani, 2014). An environment is often implemented as a software component, whose state is considered as the state of the environment. The operations that allow the interaction with this software component are used to implement, on one hand, the effect of actions that agents can perform in the environment and, on the other hand, the elements of the environment that may be perceived by the agent.

A MAS may be used for various purposes. For example, to coordinate agents’ interactions by exchanging information through it (i.e., using environment as a shared space), or to provide
agents the sense and act abilities to observe and modify the environment’s state (Dastani, 2014; Parunak and Weyns, 2007; Weyns et al., 2004). In particular, an environment may provide artefacts or services to allow agents to manage their coordination or to exchange information. An environment may also provide various sense and act modalities such as blocking and non-blocking sense operations, event broadcasting or subscription mechanisms, and synchronous or asynchronous actions.

The implementation of environments therefore demands some special and sophisticated tools. Dedicated programming languages and development frameworks are necessary to allow direct and effective implementations of related concepts and abstractions of resources, services, and objects. They should also provide programming constructs to implement artefacts, processes, and several sense and action types. For this purpose, one may consider the A&A (Agent and Artefact) meta-model (Omicini et al., 2008) as a generic approach for modelling environments. In the A&A meta-model, agents are the (pro-)active entities in charge of the goals/tasks that altogether form the whole MAS’ behaviour, whereas artefacts are the reactive entities providing the services and functions that make individual agents work together, and that shape the multiagent environment according to the MAS’ needs. An example of a language based on this meta-model is CArtAgO (Ricci et al., 2009).

2.5.3 Programming Organisations

The overall behaviour of a MAS depends on the behaviour of every agent it comprises. The MAS’ objectives may be different or even conflicting with the individual agents’, although they may be ensured by controlling and coordinating agents’ behaviours and interactions individually. There have been various proposals for regulating and organising the behaviours of individual agents. Some of these proposals advocate the use of coordination artefacts that are specified in terms of low-level coordination concepts such as synchronisation (Dastani, 2014). Other approaches are motivated by organisational models, normative systems, or electronic institutions (Dastani et al., 2009; Esteva et al., 2002, 2004; Grossi, 2007; Hübner et al., 2010; Jones and Sergot, 1993). In these approaches, agents’ behaviours are regulated through norms and organisational concepts that are either used by agents deciding how to behave, or being enforced or regimented by an exogenous organisation component. An example of these languages is Moise\(^+\) (Hübner et al., 2007).

2.5.4 The JaCaMo Framework

JaCaMo is a multiagent programming platform which introduces a new programming paradigm called multiagent oriented programming (Boissier et al., 2013b). It provides high-level first-class support for the development of agents, environments, and organisations in synergy by integrating, while preserving separation of concerns, three orthogonal programming paradigms: agent-oriented, organisation-oriented, and environment-oriented. A system built with JaCaMo is given by an agent organisation programmed in Moise\(^+\) (Hübner et al., 2007), autonomous agents programmed in Jason (Bordini et al., 2007), and shared distributed artefact-based environments programmed in CArtAgO (Ricci et al., 2009) (see Figure 2.5).

Jason is a platform for the development of MAS that incorporates an agent-oriented programming language. The logic-based BDI-inspired language AgentSpeak, initially conceived by Rao (Rao, 1996), was later much extended in a series of publications so as to make it suitable as a practical agent programming language. These extensions led to the variant of AgentSpeak that was made available in Jason (Bordini et al., 2007).

CArtAgO (Ricci et al., 2009) is a framework and infrastructure for environment programming (Ricci et al., 2011) and execution in a MAS. The underlying idea is that the environment can be used as a first-class abstraction for designing MAS, as a computational layer encapsulating functionalities and services that agents can explore at runtime (Weyns et al., 2007). As they are based on the A&A meta-model (Omicini et al., 2008), in CArtAgO such software environments may be designed and programmed as a dynamic set of computational entities called artefacts, collected into workspaces, possibly distributed among various nodes of a network.
Finally, the MoiSe+ framework (Hubner et al., 2007) implements a programming model for the organisational dimension. This approach includes an organisation modelling language, an organisation management infrastructure (Hübner et al., 2010), and support for organisation-based reasoning mechanisms at the agent level (Hubner et al., 2007).

JaCaMo integrates these three platforms by defining a semantic link among concepts of the different programming dimensions – agent, environment, and organisation – at the meta-model and programming levels, in order to obtain a uniform and consistent programming model aimed at simplifying the combination of those dimensions when programming MAS.

2.6 Discussion

Table 2.1, adapted from Nardin (2015), summarises which of the requirements pointed out in Section 1.1 are fulfilled by the main available sanctioning enforcement frameworks. As can be seen, none of them fully tackles the four requirements proposed by the author. Therefore, they do not support different types of sanctions and provide agents with sanction reasoning and decision capabilities.
This work presents a sanctioning norm enforcement framework that fulfils all the foregoing requirements through the use of various contextual factors, as explained in the next two chapters. Building upon the sanctioning process described in Section 2.3, this framework allows agents societies to function in a much more similar fashion as humans societies. To demonstrate such framework in practice, Chapter 4 covers a series of experiments wherein agents dispose of formal (norm enforcement) and informal (trust and reputation) sanctions as a regulation framework. As prospect theory is a descriptive model which tries to model real-life choices, rather than optimal decisions, agents in the experiments base their decisions on such model. These experiments were developed in JaCaMo not only due to the separation of concerns it provides in terms of agents, artefacts, and organisations implementation, but also due to its popularity among the MAS community.
Chapter 3

GAVEL Framework

GAVEL is an adaptive sanctioning enforcement framework based on the conceptual sanctioning process model introduced in Section 2.3. It enables agents to decide for the most appropriate sanctions to apply depending on their current context assessed by a set of sanctioning decision factors.

This chapter aims to provide a thorough description of GAVEL in a bottom-up approach. Firstly, it introduces and formalises the elements which compose the enforcement model in Section 3.1. Secondly, Section 3.2 describes technical aspects of the framework as well as an integration with JaCaMo called GAVEL for JaCaMo. Finally, the chapter ends with a discussion on how GAVEL tackles the four requirements raised in Chapter 1, its benefits, and shortcomings.

3.1 Formal Definition

In this section, we define formally and refine the conceptual sanctioning process model, proposed by Nardin et al. (2016), and presented in Section 2.3. Both norm violation and compliance are considered in the process, respecting the general notion of sanction as a negative or positive reaction to normative behaviours. The entire sanctioning process is realised by agents endowed with special capabilities (i.e., Detector, Evaluator, Executor, Controller, and Legislator) supported by specialised data repositories (De Jure and De Facto). All components of the norm enforcement framework are defined next.

3.1.1 NMAS

Definition 1. (NMAS) A NMAS is a system composed of a set of autonomous and heterogeneous agents situated in a shared environment, whose actions and interactions are ruled by norms and sanctions. A NMAS is defined as

\[ \text{NMAS} = \langle \mathcal{E}_{nv}, \mathcal{A}_g, \mathcal{R}, \mathcal{A}_c, \mathcal{N}, \mathcal{S}, \mathcal{L} \rangle, \]

where

- $\mathcal{E}_{nv}$ is the environment that may assume any of a finite set of discrete states;
- $\mathcal{A}_g = \{ag_i : i \leq |\mathcal{A}_g|\}$ is the set of agents that can act on the environment or interact;
- $\mathcal{R} = \{r_i : i \leq |\mathcal{R}|\}$ is the set of domain application roles that agents can play;
- $\mathcal{A}_c = \{\alpha_i : i \leq |\mathcal{A}_c|\}$ is the set of actions that agents can perform;
- $\mathcal{N} = \{n_i : i \leq |\mathcal{N}|\}$ is the set of norms prescribing the agents’ behaviours;
- $\mathcal{S} = \{s_i : i \leq |\mathcal{S}|\}$ is the set of sanctions prescribing possible reactions to norm violation or compliance;
- $\mathcal{L} = \mathcal{N} \times \mathcal{S}$ is the set of links (i.e., associations) between norms and sanctions.
3.1.2 Norms, Sanctions and Links

**Definition 2.** (*Norm*) A norm \( n_i \in N \) is a guide of conduct prescribing how agents ought to behave in a given situation. A norm is defined as

\[
n_i = (\text{status}, \text{activation}, \text{issuer}, \text{target}, \text{deactivation}, \text{deadline}, \text{content}),
\]

where

- \( \text{status} \in \{\text{enabled}, \text{disabled}\} \) indicates whether \( n_i \) is in force;
- \( \text{activation} \) is the set of contextual conditions that renders the norm applicable;
- \( \text{issuer} \in \mathcal{A}g \) identifies the entity that originally issued the norm;
- \( \text{target} \in \mathcal{A}g \) identifies the agent to which the norm is addressed;
- \( \text{deactivation} \) is the set of contextual conditions that renders the norm no longer applicable once active;
- \( \text{deadline} \) is the time limit to comply with the norm;
- \( \text{content} \) is the criteria prescribing the agents’ behaviours.

**Definition 3.** (*Norm Instance*) A norm instance \( n'_i \) is the result of applying a ground substitution to a norm \( n_i \). A norm instance is defined as

\[
n'_i = (\text{status}', \text{activation}', \text{issuer}', \text{target}', \text{deactivation}', \text{deadline}', \text{content}'),
\]

where each term of \( n'_i \) unifies with its corresponding term in \( n_i \).

**Definition 4.** (*Sanction*) A sanction \( s_i \in S \) is a reaction to a norm compliance or violation. A sanction is defined as

\[
s_i = (\text{status}, \text{activation}, \text{category}, \text{content}),
\]

where

- \( \text{status} \in \{\text{enabled}, \text{disabled}\} \) indicates whether \( s_i \) is in force;
- \( \text{activation} \) is the set of contextual conditions that renders the sanction applicable;
- \( \text{category} \) is defined in accordance with the sanction typology described in Section 2.3.1, such that

\[
\text{category} = (\text{purpose}, \text{issuer}, \text{locus}, \text{mode}, \text{polarity}, \text{discernability}),
\]

where

- \( \text{purpose} \in \{\text{Punishment, Reward, Enablement, Guidance, Incapacitation}\}, \)
- \( \text{issuer} \in \{\text{Formal, Informal}\}, \)
- \( \text{locus} \in \{\text{Self-Directed, Other-Directed}\}, \)
- \( \text{mode} \in \{\text{Direct, Indirect}\}, \)
- \( \text{polarity} \in \{\text{Positive, Negative}\}, \)
- \( \text{discernability} \in \{\text{Noticeable, Unnoticeable}\}; \)
- \( \text{content} \) is the specification of the set of actions representing the sanction.

**Definition 5.** (*Sanction Instance*) A sanction instance \( s'_i \) is the result of applying a ground substitution to a sanction \( s_i \). A sanction instance is defined as
\[ s'_i = \langle\text{status}', \text{activation}', \text{category}', \text{content}'\rangle, \]

where each term of \( s'_i \) unifies with its corresponding term in \( s_i \).

**Definition 6.** (Link) A link \( l_i \in L \) is an association between a norm and a subset of sanctions. A link is defined as

\[ l_i = \langle n_i, S L_{n_i} \rangle, \]

where

- \( n_i \in N \) is the norm being associated;
- \( S L_{n_i} = \{sl_j \mid sl_j = \langle\text{status}, s_j\rangle\} \) is the set of sanction links to \( n_i \), where
  - \( \text{status} \in \{\text{enabled}, \text{disabled}\} \) indicates whether \( sl_j \) is in force and
  - \( s_j \in S \) is the sanction being associated.

An enabled link states that an agent may consider a sanction \( s_j \) as a possible reaction to the compliance or violation of the norm \( n_i \).

### 3.1.3 Repositories

The framework defines two types of data repositories: De Jure and De Facto.

**Definition 7.** (De Jure) The De Jure (\( DJ \)) stores specifications of norms and sanctions and their associations. It is defined as

\[ DJ = \langle N^{DJ}, S^{DJ}, L^{DJ} \rangle, \]

where

- \( N^{DJ} \subseteq N \) is the set of all norms stored in \( DJ \);
- \( S^{DJ} \subseteq S \) is the set of all sanctions stored in \( DJ \);
- \( L^{DJ} \subseteq L \) is the set of all links between norms and sanctions stored in \( DJ \).

**Definition 8.** (De Facto) The De Facto (\( DF \)) is a repository of historical information about sanction decisions, applications, and outcomes. It is defined as

\[ DF = \langle SD^{DF}, SA^{DF}, SO^{DF} \rangle, \]

where

- \( SD^{DF} \) (Sanction Decision Set) represents the set of sanction decisions made by Evaluators and stored in \( DF \). Each sanction decision \( sd_i \in SD^{DF} \) is defined as
  \[ sd_i = \langle\text{time}, \text{detector}, \text{evaluator}, \text{target}, n'_i, s'_i, \text{cause}\rangle, \]

  where
  - \( \text{time} \) indicates the global time at which the sanction was decided;
  - \( \text{detector} \in Ag \) identifies the agent that reported the norm compliance or violation;
  - \( \text{evaluator} \in Ag \) identifies the agent that decided the sanction;
  - \( \text{target} \in Ag \) identifies the agent to which the sanction is directed;
  - \( n'_i \) is the norm instance which was evaluated by the \( \text{evaluator} \);
  - \( s'_i \) is the sanction decided by the \( \text{evaluator} \) for the \( \text{target} \) in response to \( n'_j \);
– cause ∈ \{compliance, violation\} indicates what led the evaluator to decide for the sanction \(s_k\).

- \(\mathcal{SA}^{\mathcal{DF}}\) (Sanction Application Set) represents the set of sanction applications executed by Executors. Each sanction application \(sa_i \in \mathcal{SA}^{\mathcal{DF}}\) is defined as

\[
sa_i = (time, sd_j, executor),
\]

where

- \(time\) indicates the global time at which the sanction was applied;
- \(sd_j \in \mathcal{SD}^{\mathcal{DF}}\) is the sanction decision to which \(sa_i\) is related;
- \(executor \in \mathcal{Ag}\) identifies the agent that applied the sanction.

- \(\mathcal{SO}^{\mathcal{DF}}\) (Sanction Outcome Set) represents the set of effects which resulted from the sanction observed by a Controller. Each sanction outcome \(so_i \in \mathcal{SO}^{\mathcal{DF}}\) is defined as

\[
so_i = (time, sa_j, controller, efficacy),
\]

where

- \(time\) indicates the global time at which the efficacy of the sanction was assessed;
- \(sa_j \in \mathcal{SA}^{\mathcal{DF}}\) is the observed sanction application;
- \(controller \in \mathcal{Ag}\) identifies the agent that observed the outcome;
- \(efficacy\) indicates how effective the sanction was in promoting norm compliance. It can use discrete (e.g., effective and ineffective) or continuous (e.g., \([-1, 1]\)) values.

### 3.1.4 Capabilities

The \textit{GAVEL} framework defines five capabilities: Detector, Evaluator, Executor, Controller, and Legislator. Agents having these capabilities perform tasks in different stages of the sanctioning process.

**Definition 9.** (Detector) The Detector perceives the environment and detects a norm violation or compliance. It \textit{watches} for normative events, \textit{creates} norm instances, and \textit{reports} compliances and violations to an Evaluator. The \textit{watch} function is defined as

\[
watch : e \times \mathcal{KB} \times \mathcal{N}_{\text{enabled}} \rightarrow \mathcal{N}',
\]

where

- \(e\) is the event to be analysed;
- \(\mathcal{KB}\) is the Detector’s knowledge base;
- \(\mathcal{N}_{\text{enabled}} = \{n_i | n_i \in \mathcal{N} \land n_i.status = \text{enabled}\}\) is the set of enabled norms known by the agent;
- \(\mathcal{N}'\) is the set of norm instances of \(\mathcal{N}_{\text{enabled}}\) whose activation condition holds given \(e\) and the \(\mathcal{KB}\) \((e \land \mathcal{KB} \models n_i.activation)\).

Each norm instance \(n'_i \in \mathcal{N}'\) obtained from \textit{watch} is assessed as complied, violated, or deactivated. If \(n'_i\) was complied or violated, then the Detector reports this fact to an Evaluator.
Definition 10. \((\text{Evaluator})\) The Evaluator receives from the Detector the report of a violation or compliance of a norm instance \(n_i'\). It then obtains from the De Jure repository all the applicable sanctions associated with \(n_i'\) by enabled links to decide for the appropriate sanctions to apply, if any. This task is performed by the \textit{evaluate} function, which is defined as

\[
evaluate : n_i' \times KB \times S\mathcal{L}_{n_i', \text{enabled}} \rightarrow SD_{n_i'}^{2,\mathcal{G}},
\]

where

- \(n_i'\) is the norm instance to be evaluated;
- \(KB\) is the knowledge base from which the agent extracts contextual factors to be considered in the evaluation;
- \(S\mathcal{L}_{n_i', \text{enabled}}\) is the set of enabled sanction links associated with \(n_i'\);
- \(SD_{n_i'}^{2,\mathcal{G}}\) is the set of sanction decisions for \(n_i'\).

Definition 11. \((\text{Executor})\) The Executor agent \(ag_i\) receives from the Evaluator a sanction decision \(sd_j \in SD\) and decides whether or not to \textit{execute} it \(^1\). The \textit{execute} function maps a sanction decision received to actions in the environment.

\[
execute : sd_j \rightarrow Ac.
\]

If the actions defined in \(sd_j\) are successfully executed, then the Executor records the sanction application \(sa_k = \langle \text{time}_a, sd_j, ag_i \rangle\) in the \(SA^{2,\mathcal{G}}\).

Definition 12. \((\text{Controller})\) The Controller \(ag_i\) monitors the outcomes of a sanction application \(sa_k\) to determine its efficacy and records its judgement as a sanction outcome \(so_j = \langle \text{time}_o, sa_k, ag_i, \text{efficacy} \rangle\) in the \(SO^{2,\mathcal{G}}\).

Definition 13. \((\text{Legislator})\) The Legislator updates norms, sanctions, and their associations in the De Jure based on the assessment of the De Jure and the De Facto repositories along with its knowledge base.

\[
legislate : DJ \times DF \times KB \rightarrow DJ
\]

3.1.5 Sanctioning Process

Figure 3.1 presents a sequence of actions performed by the agents endowed with the capabilities defined in the \textit{GAVEL} framework. This figure illustrates \textit{GAVEL}'s sanctioning process as a sequence diagram as an extension and improvement upon Nardin \textit{et al.}'s (2016) sanctioning process shown in Figure 2.2. The sequence is composed of the following steps:

1. The Legislator adds norms and sanctions specifications into the De Jure repository;
2. The Detector becomes aware of the norms stored in the De Jure and begins watching for normative events in the environment;
3. A Normative Actor produces a normative event \(e_1\) ruled by the norm \(n\);
4. The Detector processes \(e_1\), creates a norm instance \(n'\) out of it, and observes \(n'\) until it is either complied, violated, or deactivated;
5. After noticing that the event produced by the Normative Actor violates or complies with \(n'\), the Detector reports \(n'\) to the Evaluator;

\(^1\)An Executor which refuses to apply a sanction might as well face second-order sanctions.
6. The Evaluator processes \( n' \), decides for a sanction \( s \) based on De Jure and De Facto, creates an instance \( s' \) of \( s \), registers the decision \( sd \) with \( s' \) as the chosen sanction for \( e_1 \) in De Facto, registers the sanction decision in \( S_\mathcal{D}^{\mathcal{D}_F} \) and transmits such decision to an Executor;

7. The Executor applies \( s' \) against the Normative Actor and registers such application \( sa \) in \( S_\mathcal{A}^{\mathcal{D}_F} \);

8. The Controller notices the application of \( sa \) and begins to monitor the environment to determine the efficacy of \( sa \);

9. The Normative Actor produces another normative event \( e_2 \) ruled by the same norm \( n \);

10. The Controller notices \( e_2 \), determines the efficacy of \( sa \), and registers the outcome \( so \) in \( S_\mathcal{E}^{\mathcal{D}_F} \).

![Figure 3.1: Sanctioning process.](image)

This sequence of events may happen multiple times as the system is running. For every normative event generated there will be a similar sequence, possibly only differing by context-related data.

The following section describes how not only this process but all the other model concepts have been implemented in GAVEL.

### 3.2 Implementation

The GAVEL framework\(^2\) has been implemented in Java and is mainly divided into three packages (see Figure 3.2):

- **gavel.api** provides interfaces for all the elements of the model;
- **gavel.base** provides abstract classes which can be used as basis for customisation of some elements of the model;
- Source code available at [https://github.com/gavelproject/gavel](https://github.com/gavelproject/gavel).

\(^2\)Source code available at [https://github.com/gavelproject/gavel](https://github.com/gavelproject/gavel).
3.3 IMPLEMENTATION

Figure 3.2: GAVEL’s architecture.

gavel.impl contains generic concrete implementations of the model elements (e.g., norms, sanctions, norm-sanction links, and data repositories) according to contracts prescribed by interfaces defined in gavel.api and provides utility classes with factory, parsing, and other supporting methods.

The framework includes generic in-memory data storage implementations for three data repositories:

**DeJure** stores and provides operations to manage norms, sanctions and norm-sanction links. At runtime, these elements can be created, retrieved, updated, or deleted;

**DeFacto** stores sanction decisions, applications, and outcomes at runtime;

**CapabilityBoard** stores capability assignment rules and the capabilities possessed by agents. If an agent has a certain capability and the repository is informed, then such information will be available for the entire system.

It is often desirable to specify norms, sanctions, norm-sanction links, and capability assignment rules before the system starts running. Thus, system designers can provide two specification files using XML (eXtensible Markup Language):

- **Regulative specification** — defines norms, sanctions, and norm-sanction links specifications that will be initially stored in DeJure, as demonstrated in Algorithm 1.

- **Capability assignment specification** — defines rules specifying which agents will be allowed to possess which of the capabilities defined in the model, as demonstrated in Algorithm 2.

Notice that GAVEL does not provide execution plans for each capability as these are dependent on the MAS platform.

In addition to the standard Java implementation, another contribution of this work is GAVEL for JaCaMo\(^3\). This is a reusable framework which integrates GAVEL with JaCaMo (Boissier et al., 2016), a widely used MAS platform. The benefit of such integration is twofold: (i) GAVEL repositories are provided as CArtAgO (Ricci et al., 2009) artefacts that may be used by agents; and (ii) since Jason supports meta-programming (Bordini et al., 2007), agents may acquire plans at runtime from CapabilityBoard to learn how to perform tasks inherent to any of the sanctioning capabilities (Detector, Evaluator, Executor, Controller, Legislator).

\(^3\)Source code available at https://github.com/gavelproject/gavel-jacamo.
Algorithm 1 Regulative specification example

```xml
<?xml version="1.0" encoding="utf-8"?>
<regulative-spec>
  <norms>
    <norm id="n1" status="enabled">
      <activation>simulation("running")</activation>
      <issuer>Alice</issuer>
      <target>Player</target>
      <deactivation>false</deactivation>
      <deadline>simulation("finished")</deadline>
      <content>obligation (author ([Player|_], Paper)</content>
    </norm>
  </norms>
  <sanctions>
    <sanction id="s1" status="enabled">
      <activation>period(morning) &amp; not .my_name(Target)</activation>
      <category>
        <purpose>punishment</purpose>
        <issuer>informal</issuer>
        <locus>other_directed</locus>
        <mode>indirect</mode>
        <polarity>negative</polarity>
        <discernability>unnoticeable</discernability>
      </category>
      <content>gossip (Target)</content>
    </sanction>
  </sanctions>
  <ns-links>
    <ns-link status="enabled">
      <nid>n1</nid>
      <sid>s1</sid>
    </ns-link>
  </ns-links>
</regulative-spec>
```

### 3.3 Discussion

Recalling from Chapter 1, there are four requirements that norm enforcement frameworks must address in order to support multiple sanction types and endow agents with reasoning and decision capabilities (Nardin, 2015):

**R1** Support for multiple categories of sanctions;

**R2** Potential association of multiple sanctions with a norm violation or compliance;

**R3** Reasoning about the most adequate sanction to apply depending on different factors; and

**R4** Adaption of the sanction content depending on context.

First, **GAVEL** covers a comprehensive sanction typology of six dimensions, proposed by Nardin et al. (2016) and presented in Section 2.3.1. The field *category* from the **GAVEL**’s definition of sanction is where such dimensions go into. The support for multiple categories of sanctions (R1) is a result from the various combinations of values, across the different sanction typology’s dimensions, stored in the *category* field.

Second, multiple sanctions might be easily associated with a norm violation or compliance (R2) through *links*. By definition, norms and sanctions are completely decoupled in **GAVEL**, being *links* the elements which connect them. There is no limit on the number of *links* associating a norm to a sanction and vice-versa. Because detector, evaluator, and executor agents may deal with the possibly
different sanctions triggered as a result from a norm compliance or violation, different sanctions may also be applied accordingly.

Third, GAVEL is very open about the application of sanctions. Agents endowed with the evaluator capability are free to decide for the most appropriate sanctions, if any, to apply in response to a norm compliance or violation. Such decision may be part of a reasoning process the evaluator agent may conduct probably based on contextual factors (R3).

The fourth and final requirement (R4) is accomplished mainly via the sanction’s content field. Using logical formulas, this field specifies the actions which must be carried representing the sanction. These formulas may as well contain different variables and mathematical expressions to be evaluated by agents during runtime, thus making the sanction content adaptive.

Furthermore, other advantages for using GAVEL are its flexibility and adaptability. GAVEL can be treated as a component which can be connected to or implemented within any agent. By using it, capable agents are also free to update the legislation (De Jure) in order to obtain higher levels of norm compliance. As these decisions are all dependent on context and historical facts, GAVEL can, therefore, assure high level of flexibility and adaptability for norm enforcement in NMAS.

Conversely, GAVEL’s main disadvantages are limited control and predictability of final results. These are actually direct consequences of the flexibility it provides. As the sanctioning framework depends on the system’s history and evolution, this influences how agents will learn and apply sanctions.

Algorithm 2 Capability specification example

```xml
<capability-assignment>
  <detector>
    <agent>Alice</agent>
    <agent>Bob</agent>
    <agent>Carol</agent>
  </detector>
  <evaluator>
    <agent>Alice</agent>
    <agent>Bob</agent>
  </evaluator>
  <executor>
    <agent>Alice</agent>
  </executor>
  <controller>
    <agent>Alice</agent>
  </controller>
  <legislator>
    <agent>Dan</agent>
  </legislator>
</capability-assignment>
```
Chapter 4

Case Study

This chapter presents a case study in the domain of Economics. The GAVEL for JaCaMo framework, shown in Chapter 3, has been used to implement a version of the Public Goods Game (PGG) (Ledyard, 1995) partially inspired by Giardini et al. (2014). Several experiments have been conducted to evaluate GAVEL as a framework to enforce norms through sanctions, thus promoting cooperation. Section 4.1 describes the PGG model and its dynamics. The implementation using GAVEL for JaCaMo is described in Section 4.2. Then, Section 4.3 presents the experiments that have been conducted and the obtained results, whose analysis is provided in Section 4.4.

4.1 Public Goods Game (PGG)

Broadly used in experimental economics, agents in the PGG have private tokens and secretly choose whether to contribute to a public pool. The tokens in this pool are multiplied by a benefit factor and evenly divided among players. In a typical set-up in experimental economics an experimenter endows, for example, six players with $10 each. The players are then offered to invest their money into a common pool knowing that the experimenter will triple the amount in the pool and distribute it equally among all participants irrespective of their contributions. If all players cooperate and contribute their $10, they will end up with $30 each. However, each player faces the temptation to defect and free-ride on the other players’ contributions since each invested dollar yields only a return of 50 cents to the investor\(^1\). Therefore, the “rational” and dominating solution is to defect and invest nothing. As a result, groups of rational players will forego the public good and are thus unable to increase their initial endowment. This leads to a deadlock in a state of mutual defection and economic stalemate (Hauert, 2006).

Such public goods interactions are abundant in human and animal societies. Consider for example predator inspection behaviour, alarm calls and group defence as well as health insurance, public transportation, the fight against crime or environmental issues, to name only a few (Szabó and Hauert, 2002). Fortunately, undermining the basic rationality assumptions in economics, human subjects do not always follow the rational reasoning and, of course, fare much better by doing so. From a theoretical viewpoint, the reasons for this outcome are not fully understood but likely involve issues related to voluntary interactions or reward, punishment and reputation.

4.1.1 Overview

The PGG model used in this work divides agents into groups to play the game and allows them to use either punishment or reputation as sanction strategies. The dynamics involved is structured in Algorithm 3. At each round, agents are randomly grouped (line 3) and they decide whether to free-ride or contribute a fixed amount to the public pool (line 4). The sum of the contributions in each group is multiplied by a benefit factor and evenly divided among the group members regardless

\(^1\)As every contribution is tripled and then divided equally by all six players, then, for each dollar invested, the return is \(1 \times 3 \div 6 = 50\) cents.
of their contribution (line 5). Next, the agents’ decisions are disclosed to all other agents in their group (line 6) which individually decide whether or not to apply sanctions to other agents in their group (line 7). Once sanctions are applied (line 8), agents with less than zero tokens are eliminated from the game (lines 9–13).

Algorithm 3 Public Goods Game main cycle

```
1: Initialise agents
2: for number of rounds do
3:   Random group formation
4:   Agents make their contribution decision
5:   Gather and distribution of contributions in each group
6:   Disclose contribution decisions in the group
7:   Agents make their sanction decisions
8:   Apply sanctions
9:   for each agent do
10:      if Agent’s tokens < 0 then
11:         Agent is culled from the game
12:      end if
13:   end for
14: end for
```

Notice that there are two steps in Algorithm 3 for which agents should have specific decision strategies: contribution decision (line 4) and sanctioning decision (line 7). After joining a group, agents first decide whether to contribute based on their individual contribution strategy. Then, after contribution decisions of other members in the group are disclosed, agents that have contributed at the current round may decide whether to sanction free-riders and, if so, which sanction to apply. These strategies are described next.

4.1.2 Contribution Strategies

Agents may have one of four types of contribution strategies:

- **Cooperator** (C) who always contributes to the public pool and does not sanction other agents;
- **Free-Rider** (FR) who never contributes to the public pool and does not sanction other agents;
- **Utilitarian** (U) who contributes based on decision factors and prospect theory.

An utilitarian agent $ag_i$ recognises an agent $ag_j$ as a free-rider if the trust score that $ag_i$ has on $ag_j$ is below a certain threshold. Agents keep an individual record of all other agents in the game. Each agent $ag_i$ calculates the trust score on $ag_j$ by taking the weighted average of their direct experience and the reputational information received from others about $ag_j$, which is defined in Equation 4.1.

$$T_{ij} = W \times \Delta E + (1 - W) \times \Delta I,$$

In the latter, $T_{ij} \in [0, 1]$ is the trust score the agent $ag_i$ has on agent $ag_j$ (where 0 and 1 respectively means the lowest and highest trust value), $\Delta E$ is the proportion of good personal experiences $ag_i$ had with $ag_j$, $\Delta I$ is the average reputational information received about $ag_j$, and $W$ is the weight given to the personal experiences.

Utilitarian agents’ decisions about whether to contribute are based on the prospect theory’s principles, as described in Section 2.4.2. In order to calculate the utility of contribution strategy, the agent uses a reference-dependent model proposed by Kőszegi and Rabin (2006, 2007, 2009).

The variables involved in the utility computation are the following:

- $M_{ij}(t)$ is the set of team mates of agent $i$ at round $t$;
• $M_{ij}^F(t)$ is the set of team mates of agent $i$ at round $t$ considered as free-riders;
• $D_{ij}$ is the set of defections that agent $i$ made as a team mate of agent $j$;
• $P_{ji}$ is the set of punishments inflicted by agent $j$ on agent $i$;
• $\beta$ is the benefit factor;
• $c$ is the cooperation cost;
• $\psi$ is the punishment cost.

In this contribution strategy model, agents derive utility from gains and losses. First, let us define the function used to calculate revenue. If an agent $i$ cooperates, then its revenue at round $t$, if everyone else also cooperates, is defined according to Equation 4.2.

\[ r_i(t|\text{coop}) = \beta \times c. \]  \hspace{1cm} (4.2)

Otherwise, if agent $i$ does not cooperate but every one else does, its revenue for defecting is the one defined in Equation 4.3.

\[ r_i(t|\text{def}) = \beta \times c \times |M_i(t)| + 1 \]  \hspace{1cm} (4.3)

At first glance, it seems that it is always better to contribute rather than defecting. However, the losses, or expenses, involved in every decision should also be considered as they also differ depending on the contribution strategy. The expense function for cooperation is simply the cooperation value invested by agent $i$, as defined in Equation 4.4.

\[ \chi_i(t|\text{coop}) = c \]  \hspace{1cm} (4.4)

Conversely, since agents might suffer material sanctions as a result of defections, Equation 4.5 defines the expense function for defection considering both the cost and a likelihood estimation of a given defector being punished at round $t$.

\[ \chi_i(t|\text{def}) = |M_i(t)| \times \psi \times \omega_i(t) \]  \hspace{1cm} (4.5)

The function $\omega_i(t)$ represents agent $i$’s estimation of the average probability of being punished by team mates at round $t$, regardless of sanctioning strategy, and is defined by Equation 4.6. Notice that the powers $0^{D_{ij}}$ used to calculate $\omega_i(t)$ serve the purpose of estimating probability 0.5 when two agents have never played before.

\[ \omega_i(t) = \frac{1}{|M_i(t)|} \sum_{j \in M_i(t)} \frac{|P_{ji}| + 0^{D_{ij}}}{|D_{ij}| + 2 \times 0^{D_{ij}}} \]  \hspace{1cm} (4.6)

Finally, let us define the utility functions. The utility to cooperate at round $t$ is defined by Equation 4.7, where $\eta > 0$ is the weight the agent $i$ attaches to gain-loss utility, and $\lambda > 1$ is the coefficient of loss aversion.

\[ \mu(t|\text{coop}) = r_i(t|\text{coop}) - \chi(t|\text{coop}) + q_i(t|\text{coop}) \times (\eta \times (r_i(t|\text{coop}) - r_i(t|\text{def})) - \eta \times \lambda \times \chi_i(t|\text{coop})) \]  \hspace{1cm} (4.7)

The first line of Equation 4.7 is the consumption utility from cooperating. The second line is the gain-loss utility comparing cooperating with not cooperating, which is assessed as a gain of $r_i(t|\text{coop}) - r_i(t|\text{def})$ in income and a loss of $\chi_i(t|\text{coop})$ in expense. The function $q_i(t|s)$ represents the proportion of agents considered cooperators in the agent $i$’s group at round $t$ given strategy $s$,
as defined in Equation 4.8.

\[
q_i(t|s) = \begin{cases} 
\frac{|M_{Fr}^i(t)|}{|M_i(t)|}, & \text{if } s = \text{coop} \\
1 - \frac{|M_{Fr}^i(t)| + 1}{|M_i(t)|}, & \text{otherwise.}
\end{cases}
\]  

(4.8)

The utility from defecting is defined by the Equation 4.9. Like in Equation 4.7, the first line of Equation 4.9 is the consumption utility from defecting. The second line is the gain-loss utility comparing not defecting with cooperating, which is assessed as a gain of \(\chi_i(t|\text{coop}) - \chi_i(t|\text{def})\) in income and a loss of \(r_i(t|\text{coop}) - r_i(t|\text{def})\) in expense.

\[
\mu(t|\text{def}) = r_i(t|\text{def}) - \chi(t|\text{def}) + (1 - q_i(t|\text{coop})) \times (\eta \times (\chi_i(t|\text{coop}) - \chi_i(t|\text{def})) - \eta \times (r_i(t|\text{coop}) - r_i(t|\text{def})))
\]

(4.9)

After an agent calculates both utilities, \(\mu(t|\text{coop})\) and \(\mu(t|\text{def})\), it takes both numbers into a vector \(z = \{\mu(t|\text{coop}); \mu(t|\text{def})\}\) and normalises them into a probability distribution using the softmax function, \(\sigma(z)\). Then, the agent randomly picks a number \(n\) from a uniform distribution between 0 and 1. If \(n \geq \sigma(z)\), the agent cooperates. Otherwise, it defects.

4.1.3 Sanctioning Strategies

For this model, only utilitarian agents may sanction other agents. They use features from GAVEL to guide their sanction choice towards free-riders. For each experiment, the sanction strategy adopted by utilitarian agents is one of the following:

- **Punishment** (P): Agents simply sanction others directly by inflicting a fine (i.e., punishment cost), a material sanction for which they must pay a smaller fee (i.e., enforcement cost);

- **Mixed** (M): Agents decide whether to either sanction indirectly using gossip (i.e., spread a bad reputation) or directly using punishment by comparing the free-rider’s trust score with a randomly picked number from a uniform distribution between 0 and 1. If the free-rider’s trust score is less than the random number, the agent punishes the free-rider, otherwise it gossips.

Each agent has a limit on the number of sanctions it may apply at each round, whether it is gossip or punishment. If this limit is reached, sanction applications are no longer possible for the current round. Each punishment inflicted on free-riders has a cost to the agent inflicting it, thus an agent can only sanction if it can afford. In behavioural economic experiments, a direct material sanction is expressed by paying a fee to reduce the payoff of another agent by a larger amount. This is also known as **costly punishment**, because while punishment is costly to the punisher, it is even more costly to the punished (Henrich et al., 2006).

4.2 Implementation

For the PGG simulation, two types of agents have been implemented: game **manager** and **player**. The manager is responsible for (1) creating rounds; (2) defining groups; (3) gathering contributions; (4) multiplying the total contribution by a benefit factor; (5) dividing the result evenly among players; and (6) disclosing the contribution of each player. Conversely, players are limited to (1) contributing to the pool; and (2) sanctioning other players based on the content of De Jure and their sanctioning strategy.

Figure 4.1 depicts a fully norm-compliant round of the game. Once the manager opens the round, players start contributing to the pool. When all contributions were made, the manager applies the benefit factor, gathers the resulting amount, distributes the proportion of each agent, discloses contributions, and closes the round.
Figure 4.1: Round with all players complying with the norm.

Figure 4.2 illustrates a round in which a sanctioning occurs. After the manager discloses the contributions, the player $p1$ notices that $p2$ did not contribute. Then, player $p1$ decides for a sanction and applies it to the player $p2$.

The pools to which players cooperate are controlled by the manager and implemented as domain artefacts using CArtAgO. All players are endowed with the Detector, Evaluator, Executor, and Controller capabilities, but for the sake of simplicity, no Legislator was included.

Only one norm regulates the players' behaviours in the PGG. As shown in Algorithm 4, this norm, identified as positive_contribution, states that every player is obliged to contribute with 1 token to every pool it participates. If an agent does not comply with the norm before the pool finishes, then the norm is violated.

Algorithm 4 Positive contribution norm

```java
1 norm(
2   id(positive_contribution),
3   status(enabled),
4   activation(
5     pool_member( Player )
6   ),
7     issuer( manager ),
8     target( Player ),
9     deactivation( false ),
10     deadline(
11     pool_status("FINISHED")
12   ),
13   content(
14     obligation(
15       contribution( Player,1)
16     )
17   )
18 )
```

The two links shown in Algorithm 5 state that the positive_contribution norm is linked to two sanctions, punishment and gossip.

Algorithm 5 Norm-sanction links

```java
1 nslink(
2   status(enabled),
3   norm_id(positive_contribution),
4   sanction_id(punishment)
5 ),
6 nslink(
7   status(enabled),
8   norm_id(positive_contribution),
9   sanction_id(gossip)
10 )
```
The punishment sanction is only applicable if the player evaluating the violation has not exceeded the limit of sanctions in the current round, is not the target, and can afford the sanction. As shown in Algorithm 6, this negative informal sanction counts as applied when the Executor directly punishes the target inflicting a pre-established cost.

**Algorithm 6 Punishment sanction**

```plaintext
1 sanction(
2     id (punishment),
3     status (enabled),
4     activation (not sanctions_credit(0)
5          & not .my_name(Target)
6          & cost_to_punish(Cost)
7          & tokens(Tokens)
8          & Cost <= Tokens
9     ),
10    category (purpose(punishment),
11        issuer(informal),
12        locus(other_directed),
13        mode(direct),
14        polarity(negative),
15        discernability(noticeable)
16     ),
17    content(punish(Target))
18 )
```

Conversely, the gossip sanction is only applicable if there is a player in another group to whom it may gossip, the player evaluating the violation has not exceeded the limit of sanctions in the current round, and the evaluator itself is not the target. As shown in Algorithm 7, this is a negative informal sanction that counts as applied when the Executor transmits reputational information about the target to an agent from another group.

**Algorithm 7 Gossip sanction**

```plaintext
1 sanction(
2     id (gossip),
3     status (enabled),
4     activation (not players_in_other_groups([])
5          & not sanctions_credit(0)
6          & not .my_name(Target)
7     ),
8    category (purpose(punishment),
9        issuer(informal),
10       locus(other_directed),
11       mode(indirect),
12       polarity(negative),
13       discernability(unnoticeable)
14     ),
15    content (gossip(Target, Reputation))
16 )
```

4.3 Experiments

A set of 16 experiments was conducted to analyse how GAVEL enables agents to reason about sanction decisions. These experiments were divided into 2 groups, one for each sanctioning strategy (i.e., punishment and mixed) employed by all agents. For each experiment, the agents population was formed by 300 agents, 100 of each using a different contribution strategy (i.e., Cooperator, Free-rider, and Utilitarian). Similarly to parameters used by Giardini et al. (2014, 2015), groups of at most 25 agents were formed in each round, the contribution to the public pool was set to 1 token, and the benefit factor was set to 3. Each agent was endowed with an initial amount of tokens to be used to contribute to the public pool or to sanction others, hereafter called *endowment*. Utilitarian agents decide whether to contribute in a certain round following a prospect theory model which considers own wealth, loss aversion factor, and estimated cooperation and defection utility based on team mates trust. An agent is considered a free-rider if the trust on it is below a threshold of 0.6.

Both punishments and gossips applications have some limiting factors. A punishment involves a cost to the punisher (i.e., enforcement cost) and a higher cost to the punished agent (i.e., punishment cost). A gossip, conversely, does not incur any cost. Moreover, it has also been established a maximum sanctioning rate for both sanction type. The maximum sanctioning rate picked is 0.75, meaning that utilitarian agents may only sanction at most 75% of their team mates in a certain round. That is, in a group of 25 agents, an agent $ag_i$ has 24 team mates, thus at most $24 \times 0.75 = 18$ agents may be sanctioned by $ag_i$ in that round. The sanctioning rate of 0.75 has been found to be a reasonable rate as it is not extreme and still allows to study the effects of multiple sanctions.
4.3.1 Description

For each scenario, the PGG model was run for 100 rounds, repeated 10 times with different random seeds for each combination of parameter values shown in Table 4.1. There are 3 parameters, 2 possible values each (low and high), thus \(2^3 = 8\) parameter combinations, aside from the sanctioning strategy.

| Parameter         | Values |
|-------------------|--------|
| Endowment         | Low 12.5 | High 50 |
| Enforcement Cost  | Low 0.05 | High 0.25 |
| Punishment Cost   | Low 0.5 | High 1.25 |

Table 4.1: Simulation parameters.

Table 4.2 contains an identification of each experiment along with parameter values used. The endowment, enforcement cost, and punishment cost are a linear transformation of the parameters used in a previous work by Giardini et al. (2015). Each parameter may be set to either a low or a high value, namely low and high endowment (Lt and Ht), low and high enforcement cost (Le and He), and low and high punishment cost (Lp and Hp). Each experiment has been labelled using the initial letter of the group’s name, namely M for mixed and P for punishment sanction, followed by a short identifier for each parameter.

| # | ID | Group | Endowment | Enforcement Cost | Punishment Cost |
|---|----|-------|-----------|------------------|-----------------|
| 1 | P-LtLeLp | Punishment | 12.5 | 0.05 | 0.5 |
| 2 | P-LtLeHp | Punishment | 12.5 | 0.05 | 1.25 |
| 3 | P-LtHeLp | Punishment | 12.5 | 0.25 | 0.5 |
| 4 | P-LtHeHp | Punishment | 12.5 | 0.25 | 1.25 |
| 5 | P-HtLeLp | Punishment | 50 | 0.05 | 0.5 |
| 6 | P-HtLeHp | Punishment | 50 | 0.05 | 1.25 |
| 7 | P-HtHeLp | Punishment | 50 | 0.25 | 0.5 |
| 8 | P-HtHeHp | Punishment | 50 | 0.25 | 1.25 |
| 9 | M-LtLeLp | Mixed | 12.5 | 0.05 | 0.5 |
| 10 | M-LtLeHp | Mixed | 12.5 | 0.05 | 1.25 |
| 11 | M-LtHeLp | Mixed | 12.5 | 0.25 | 0.5 |
| 12 | M-LtHeHp | Mixed | 12.5 | 0.25 | 1.25 |
| 13 | M-HtLeLp | Mixed | 50 | 0.05 | 0.5 |
| 14 | M-HtLeHp | Mixed | 50 | 0.05 | 1.25 |
| 15 | M-HtHeLp | Mixed | 50 | 0.25 | 0.5 |
| 16 | M-HtHeHp | Mixed | 50 | 0.25 | 1.25 |

Table 4.2: Combinations of parameters used in the PGG experiments.

There are four main data types which were collected from the experiments in order to investigate and compare both punishment and mixed experimental groups. The following sections report the results on cooperation rate, number of sanction applications, number of survivors, and wealth. Each of which is briefly presented with charts to help understand general effects.

4.3.2 Cooperation rate results

The cooperation rate is calculated as the average proportion of cooperation per group measured, which can be formulated as in Equation 4.10, where \(r\) is the round, \(Ag^r\) is the set of agents alive in
4.3 Sanction application results

Trying to ensure social order, agents apply a large number of sanctions to defectors in each round. Figures 4.5 and 4.6 show the average number of material sanctions applied in each round of the simulations in the punishment and mixed experimental groups, respectively. Notice that, at the
beginning, when the cooperation rate is low, the number of sanctions is generally high, but after 10 to 15 rounds, when the cooperation rate is high, the number of sanctions decreases substantially.

Figure 4.5: Number of material sanctions applied over the rounds in the punishment experimental group.

Figure 4.6: Number of material sanctions applied over the rounds in the mixed experimental group.

Figure 4.7: Number of gossips made over the rounds in the mixed experimental group.
In contrast, as shown in Figure 4.7, the number of gossips transmitted in the mixed group increases over time. As members of a society of high cooperation rates are likely to have good reputation, this explains why the number of gossips increases as the number of material sanctions decreases. This reflects the fact that, in the mixed sanctioning strategy, agents tend to use lighter sanctions when defectors have a fairly good reputation.

Figures 4.8 and 4.9 give a different view on the number of sanctions applied. Differently from the previous figures, these provide the average number of sanctions applied in the whole simulation, instead of in each round. The former shows the number of material sanctions in the punishment group, whereas the latter shows the number of gossips and material sanctions in the mixed group.

![Figure 4.8: Average number of sanctions applied in each experiment of the punishment group.](image)

![Figure 4.9: Average number of sanctions applied in each experiment of the mixed group.](image)

### 4.3.4 Surviving agent results

The average number of survivors is also an important metric to keep track of in order to understand the dynamics of the cooperation rate over time. Figures 4.10 and 4.11 show the average number of survivor agents\(^2\) in each experimental group, punishment and mixed, respectively. All curves start at 300 in population size and after a steep decline in less than 15 rounds they come to an equilibrium. It is important to notice that both figures differ considerably for the experiments LtLeHp and LtHeHp as the number of survivors in the experiments with mixed strategy is greater than that in the punishment group.

---

\(^2\)This work refers to agents which have not been culled from the game, and therefore are still playing, as survivor agents.
4.3.5 Wealth results

In both experimental groups, the average accumulated wealth over time has a linear increase after approximately 10 rounds. Figures 4.12 and 4.13 illustrates how the average population wealth evolve in the punishment and mixed experimental groups, respectively. As expected, the 4 curves which represent the experiments with high endowment (He) start at 50 tokens and the other 4 with low endowment (Le) start at 12.5 due to the initial setup parameters. The first 10 rounds present a notable variation in wealth levels due to the first effects of sanctions in cooperation rate and number of survivors shown previously. Moreover, it is possible to see that the mixed group has reached higher wealth levels (near 200) compared to the punishment group.
### 4.3.6 Statistical Analysis

The following sections present statistical analyses of the data collected from the experiments. They mainly focus on comparing last results’ data distributions from both the punishment and mixed experimental groups to discover whether these strategies differ in terms of end results across different dimensions: cooperation rate, number of material sanctions, and wealth. It is important to notice that only data from the last round is used in the analysis because at this point convergence has already been reached. Let us also recall that agents from the mixed group may only use either gossip or material sanction, not both, to sanction each defector in different rounds. Because of this, the hypothesis is that both experimental groups present similar cooperation rates but the mixed group, which benefits the most from GAVEL’s features, reaches higher wealth levels, applies less material sanctions, and kills less agents.

#### Cooperation rate

The first step of this analysis is to investigate the difference between average cooperation rate found in the last round of similar experiments from both groups. In order to do this, multiple two-sided Mann-Whitney U tests (Mann and Whitney, 1947) (also known as Wilcoxon-Mann-Whitney
test) with continuity correction have been performed, one for each pair of parameter combination from both groups. The distributions of cooperation rates for each experiment may be seen in Figure 4.14 as box plots.

Figure 4.14: Box plots representing the distribution of cooperation rates for each experiment.

The tests performed check whether cooperation rates from both groups come from the same distribution. Thus, we have the following hypotheses:

Null hypothesis \((H_0) \) “the cooperation rates come from the same distribution”;

Alternative hypothesis \((H_1) \) “the cooperation rates come from different distributions”.

Before running the tests, the significance level (alpha) was set to 0.05. The results depicted in Table 4.3 show that none of the tests were statistically significant. Therefore, there is not enough evidence to support the hypothesis that cooperation rates from both groups come from different distributions, meaning the null hypothesis cannot not be discarded.

Table 4.3: Mann-Whitney \(U\) test of cooperation rates. The values in the “Punishment” and “Mixed” columns are the average cooperation rates for punishment and mixed experimental groups, respectively.

| Pair    | Punishment | Mixed   | p-value | Significance |
|---------|------------|---------|---------|--------------|
| LtLeLp  | 0.9828     | 0.9729  | 0.0885  | ns           |
| LtLeHp  | 0.9910     | 0.9880  | 0.5700  | ns           |
| LtHeLp  | 0.9748     | 0.9704  | 0.4720  | ns           |
| LtHeHp  | 0.9900     | 0.9896  | 0.9697  | ns           |
| HtLeLp  | 0.9790     | 0.9720  | 0.1699  | ns           |
| HtLeHp  | 0.9915     | 0.9905  | 0.9693  | ns           |
| HtHeLp  | 0.9825     | 0.9740  | 0.1084  | ns           |
| HtHeHp  | 0.9904     | 0.9895  | 0.3767  | ns           |

Material sanctions

Secondly, let us analyse the average number of material sanctions applied in the last round per experiment. Here, it is important to discover whether the punishment group had more material sanctions than the mixed group. To do this, let use this time multiple one-sided Mann-Whitney \(U\) tests with continuity correction, one test for each pair of parameter combination from both experimental groups. The hypotheses are the following:
Null hypothesis \((H_0)\) “the number of material sanctions come from the same distribution”;

Alternative hypothesis \((H_1)\) “the number of material sanctions in the punishment experimental group is greater than or equal to the number of material sanctions in the mixed group”.

Similarly to the analysis done for cooperation rate previously, the significance level was set to 0.05 before running the tests. The results depicted in Table 4.4 show that number of material sanctions applied in the punishment group are indeed greater. Since there is not much evidence to state that cooperation rates differ between both group, let us assume for a moment that cooperation rates are actually equivalent. If this is the case, then the mixed group has achieved equivalent levels of cooperation rate applying less material sanctions, thus saving economic resources.

Table 4.4: One-sided Mann-Whitney U test of number of material sanctions. The values in the “Punishment” and “Mixed” columns are the average number of material sanctions in the punishment and mixed experimental groups, respectively. The accepted alternative hypothesis states that the number of material sanctions in the punishment group is greater than or equal to the number of material sanctions in the mixed group.

| Pair    | Punishment | Mixed | \(p\)-value | Significance |
|---------|------------|-------|--------------|--------------|
| LtLeLp  | 15542.8    | 9712.9| 0.0001       | ***          |
| LtLeHp  | 6958.5     | 4501.6| 0.0001       | ***          |
| LtHeLp  | 14983.5    | 9582.8| 0.0001       | ***          |
| LtHeHp  | 6728.1     | 4472.2| 0.0001       | ***          |
| HtLeLp  | 25077.1    | 19475.3| 0.0001      | ***          |
| HtLeHp  | 11196.1    | 8197.5| 0.0001       | ***          |
| HtHeLp  | 25142.3    | 19445.5| 0.0001      | ***          |
| HtHeHp  | 11261.2    | 8219.1| 0.0001       | ***          |

Wealth

Finally, it has also been investigated how the population wealth varies between experimental groups. The test was performed by taking the average of the population wealth in the last round from each experiment and performed multiple two-sided Mann-Whitney U tests with continuity correction, one for each pair of parameter values combination from both experimental groups. Figure 4.15 illustrates how the averages differ in each pair. The mixed strategy seems to lead to greater wealth and this is what should be confirmed in the test. The hypotheses are the following:

Null hypothesis \((H_0)\) “the average wealth accumulated in the last round of each experimental groups come from the same distribution”;

Alternative hypothesis \((H_1)\) “the average wealth accumulated in the last round of each experimental groups does not come from the same distribution”.

The test results in Table 4.5 show that each experimental group has its own distribution of wealth. In fact, the wealth obtained in the mixed group is greater. Therefore, agents from the mixed experimental group are likely to be richer than those in the punishment group.
Figure 4.15: Bar plots illustrating the average population wealth in the end of simulations of each experiment.

Table 4.5: Mann-Whitney U test of final population wealth. The values in the “Punishment” and “Mixed” columns are the average population wealth in the last round of the punishment and mixed experimental groups, respectively.

| Pair      | Punishment       | Mixed       | p-value     | Significance |
|-----------|------------------|-------------|-------------|--------------|
| LtLeLp    | 177.656          | 192.821     | 1.83E-04    | ***          |
| LtLeHp    | 185.42           | 200.505     | 1.83E-04    | ***          |
| LtHeLp    | 163.555          | 183.768     | 1.83E-04    | ***          |
| LtHeHp    | 179.109          | 196.122     | 1.83E-04    | ***          |
| HtLeLp    | 207.498          | 221.835     | 1.83E-04    | ***          |
| HtLeHp    | 214.763          | 233.521     | 1.83E-04    | ***          |
| HtHeLp    | 182.301          | 202.639     | 1.83E-04    | ***          |
| HtHeHp    | 203.032          | 224.996     | 1.83E-04    | ***          |

4.4 Discussion

The results show that the sanction reasoning capability that GAVEL provides allowed agents from the mixed experimental group to reach cooperation rates similar to those obtained in the punishment group using less material sanctions and accumulating more wealth. In some experiments, LtLeHp and LtHeHp, the number of agents culled from the game was also considerably lower in the mixed experimental group.

Without the capability of adapting to the current context to choose a different sanction, the mixed experimental group would not present such results. By providing support for multiple categories of sanctions (i.e., gossip and material sanction), association of multiple sanctions to the violation of a single norm, reasoning about the most adequate sanction (sanction decision heuristic), and adaption of the sanction strength depending on the context (lighter vs heavier sanctions), GAVEL addresses the four requirements highlighted initially in Section 1.1. GAVEL made possible a seamless incorporation of a sanction decision heuristic, allowing agents to sanction more effectively and, specifically to the PGG, agents could accumulate more wealth over time.
Chapter 5

Conclusions and Future Work

This thesis describes the design and implementation of an adaptive sanctioning enforcement framework, called GAVEL, based on the conceptual model proposed by Nardin et al. (2016). A version of GAVEL integrated with JaCaMo, a major multi-agent system programming platform, has also been described. This version, simply called GAVEL for JaCaMo, was used to implement a version of the public goods game, a well-known social dilemma in economics wherein different sanctions may be applied in response to a norm violation. In each round, agents could decide how to contribute based on a prospect theory model and which type of sanction to use against each defector at each round of the game. The experiments were divided into two groups: punishment and mixed. The results show that the mixed sanction strategy, a simple sanctioning decision heuristic that uses reputation and punishment (material sanction), reaches cooperation rates comparable to those obtained using the punishment-only strategy, a sanctioning strategy that does not make any informed decision for sanctioning. However, greater levels of population wealth with less material sanctions could be found in the experiments using mixed sanction strategy. For some experiments, an increase on the number of survivor agents could also be noted.

The main advantages for using GAVEL are its flexibility and adaptability. GAVEL can be treated as a component which can be connected to or implemented within any agent. By using it, agents are free to choose the sanctions and intensity they deem the best to sanction a violator or complier agent. They may also update the legislation, or the De Jure repository, to obtain higher levels of norm compliance. As these decisions are all dependent on the current context and historical facts, GAVEL can, therefore, assure high level of flexibility and adaptability for norm enforcement in normative multiagent systems (NMAS).

Conversely, GAVEL’s main disadvantages are limited control and predictability of final results. These are actually direct consequences of the flexibility it provides. As the sanctioning framework depends on the system’s history and evolution, this influences how agents will learn and apply sanctions.

The next main step is to further explore the adaptability GAVEL provides. Using reinforcement learning to allow Evaluators making better sanction decisions and Legislators updating De Jure based on De Facto could lead to interesting results. Furthermore, different parameter value (e.g., group size, contribution strategy, population formation) combinations also need to be experimented so that new emergent behaviours may be found.
Appendix A

Regulative specification schema file

The schema which defines the regulative specification file is the following:

```xml
<?xml version="1.0" encoding="UTF-8"?>
<xsd:schema
    xmlns:xsd="http://www.w3.org/2001/XMLSchema"
    xmlns:gavel="https://github.com/gavelproject/gavel"
    targetNamespace="https://github.com/gavelproject/gavel"
    elementFormDefault="qualified">

<xsd:element name="regulative-spec" type="gavel:regulativeSpec" />
<xsd:complexType name="regulativeSpec">
    <xsd:sequence>
        <xsd:element name="norms" minOccurs="0" maxOccurs="1" type="gavel:norms" />
        <xsd:element name="sanctions" minOccurs="0" maxOccurs="1" type="gavel:sanctions" />
        <xsd:element name="ns-links" minOccurs="0" maxOccurs="1" type="gavel:ns-links" />
    </xsd:sequence>
</xsd:complexType>
<xsd:complexType name="norms">
    <xsd:sequence>
        <xsd:element name="norm" minOccurs="0" maxOccurs="unbounded" type="gavel:norm" />
    </xsd:sequence>
</xsd:complexType>
<xsd:complexType name="norm">
    <xsd:sequence>
        <xsd:element name="activation" minOccurs="1" maxOccurs="1" type="xsd:string" />
        <xsd:element name="issuer" minOccurs="1" maxOccurs="1" type="xsd:string" />
        <xsd:element name="target" minOccurs="1" maxOccurs="1" type="xsd:string" />
        <xsd:element name="deactivation" minOccurs="1" maxOccurs="1" type="xsd:string" />
        <xsd:element name="deadline" minOccurs="1" maxOccurs="1" type="xsd:string" />
        <xsd:element name="content" minOccurs="1" maxOccurs="1" type="xsd:string" />
    </xsd:sequence>
    <xsd:attribute name="id" type="xsd:string" use="required" />
</xsd:complexType>
```

47
<xsd:attribute name="status" type="gavel:status"
    default="enabled" />
</xsd:complexType>
<xsd:simpleType name="status">
    <xsd:restriction base="xsd:string">
        <xsd:enumeration value="enabled" />
        <xsd:enumeration value="disabled" />
    </xsd:restriction>
</xsd:simpleType>

<xsd:complexType name="sanctions">
    <xsd:sequence>
        <xsd:element name="sanction" minOccurs="0" maxOccurs="unbounded" type="gavel:sanction" />
    </xsd:sequence>
</xsd:complexType>

<xsd:complexType name="sanction">
    <xsd:sequence>
        <xsd:element name="activation" minOccurs="0" maxOccurs="1" type="xsd:string" />
        <xsd:element name="category" minOccurs="1" maxOccurs="1" type="gavel:sanctionCategory" />
        <xsd:element name="content" minOccurs="1" maxOccurs="1" type="xsd:string" />
        <xsd:attribute name="id" type="xsd:string" use="required" />
        <xsd:attribute name="status" type="gavel:status" default="enabled" />
    </xsd:sequence>
</xsd:complexType>

<xsd:complexType name="sanctionCategory">
    <xsd:sequence>
        <xsd:element name="purpose" minOccurs="1" maxOccurs="1" type="gavel:purpose" />
        <xsd:element name="issuer" minOccurs="1" maxOccurs="1" type="gavel:issuer" />
        <xsd:element name="locus" minOccurs="1" maxOccurs="1" type="gavel:locus" />
        <xsd:element name="mode" minOccurs="1" maxOccurs="1" type="gavel:mode" />
        <xsd:element name="polarity" minOccurs="1" maxOccurs="1" type="gavel:polarity" />
        <xsd:element name="discernability" minOccurs="1" maxOccurs="1" type="gavel:discernability" />
    </xsd:sequence>
</xsd:complexType>

<xsd:simpleType name="purpose">
    <xsd:restriction base="xsd:string">
        <xsd:enumeration value="punishment" />
        <xsd:enumeration value="reward" />
        <xsd:enumeration value="incapacitation" />
        <xsd:enumeration value="guidance" />
        <xsd:enumeration value="enablement" />
    </xsd:restriction>
</xsd:simpleType>

<xsd:simpleType name="issuer">
    <xsd:restriction base="xsd:string">
        <xsd:enumeration value="formal" />
        <xsd:enumeration value="informal" />
    </xsd:restriction>
</xsd:simpleType>

<xsd:simpleType name="locus">
</xsd:simpleType>
<xsd:restriction base="xsd:string">
  <xsd:enumeration value="self_directed" />
  <xsd:enumeration value="other_directed" />
</xsd:restriction>
</xsd:simpleType>
<xsd:simpleType name="mode">
  <xsd:restriction base="xsd:string">
    <xsd:enumeration value="direct" />
    <xsd:enumeration value="indirect" />
  </xsd:restriction>
</xsd:simpleType>
<xsd:simpleType name="polarity">
  <xsd:restriction base="xsd:string">
    <xsd:enumeration value="positive" />
    <xsd:enumeration value="negative" />
  </xsd:restriction>
</xsd:simpleType>
<xsd:simpleType name="discernability">
  <xsd:restriction base="xsd:string">
    <xsd:enumeration value="noticeable" />
    <xsd:enumeration value="unnoticeable" />
  </xsd:restriction>
</xsd:simpleType>
<xsd:complexType name="ns-links">
  <xsd:sequence>
    <xsd:element name="ns-link" minOccurs="1" maxOccurs="unbounded" type="gavel:nsLink" />
  </xsd:sequence>
</xsd:complexType>
<xsd:complexType name="nsLink">
  <xsd:sequence>
    <xsd:element name="nid" minOccurs="1" maxOccurs="1" type="xsd:string" />
    <xsd:element name="sid" minOccurs="1" maxOccurs="1" type="xsd:string" />
  </xsd:sequence>
  <xsd:attribute name="status" type="gavel:status" default="enabled" />
</xsd:complexType>
</xsd:schema>
Appendix B

Capabilities specification schema file

The schema which defines the capabilities specification file is the following:

```xml
<?xml version="1.0" encoding="UTF-8"?>
<xsd:schema xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:gavel="https://github.com/gavelproject/gavel"
targetNamespace="https://github.com/gavelproject/gavel"
elementFormDefault="qualified">
  <xsd:element name="capability-assignment" type="gavel:capabilityAssignment"/>
  <xsd:complexType name="capabilityAssignment">
    <xsd:sequence>
      <xsd:element name="detector" minOccurs="1" maxOccurs="1" type="gavel:capability"/>
      <xsd:element name="evaluator" minOccurs="1" maxOccurs="1" type="gavel:capability"/>
      <xsd:element name="executor" minOccurs="1" maxOccurs="1" type="gavel:capability"/>
      <xsd:element name="controller" minOccurs="1" maxOccurs="1" type="gavel:capability"/>
      <xsd:element name="legislator" minOccurs="1" maxOccurs="1" type="gavel:capability"/>
    </xsd:sequence>
  </xsd:complexType>
  <xsd:complexType name="capability">
    <xsd:sequence>
      <xsd:element name="agent" minOccurs="1" type="xsd:string"/>
    </xsd:sequence>
  </xsd:complexType>
</xsd:schema>
```
References

Abdellaoui (2000) Mohammed Abdellaoui. Parameter-free elicitation of utility and probability weighting functions. *Management science*, 46(11):1497–1512. Cited on page 12

Andrighetto and Villatoro (2011) Giulia Andrighetto and Daniel Villatoro. Beyond the carrot and stick approach to enforcement: An agent-based model. In Boicho Kokinov, Annette Karmiloff-Smith and Nancy J. Nersessian, editors, *European Perspectives on Cognitive Science*, pages 1–6, Sofia. New Bulgarian University Press. Cited on page 2

Andrighetto et al. (2013) Giulia Andrighetto, Guido Governatori, Pablo Noriega and Leendert WN van der Torre. *Normative multi-agent systems*, volume 4. Schloss Dagstuhl-Leibniz-Zentrum fuer Informatik. Cited on page 1

Aronfreed (1969) Justin Aronfreed. The concept of internalization. *Handbook of socialization theory and research*, pages 263–323. Cited on page 6

Artikis and Pitt (2008) Alexander Artikis and Jeremy V Pitt. Specifying open agent systems: A survey. In *ESAW*, pages 29–45. Springer. Cited on page 5

Barberis (2013) Nicholas C Barberis. Thirty years of prospect theory in economics: A review and assessment. *Journal of Economic Perspectives*, 27(1):173–96. Cited on page 10, 12, 13

Boella and van der Torre (2003) Guido Boella and Leendert van der Torre. Norm governed multiagent systems: The delegation of control to autonomous agents. In *IEEE/WIC International Conference on Intelligent Agent Technology, 2003. IAT 2003.*, pages 329–335. IEEE. Cited on page 5

Boella et al. (2006) Guido Boella, Leendert Van Der Torre and Harko Verhagen. Introduction to normative multiagent systems. *Computational & Mathematical Organization Theory*, 12(2-3): 71–79. Cited on page 1

Boella et al. (2008) Guido Boella, Leendert Torre and Harko Verhagen. Introduction to the special issue on normative multiagent systems. *Autonomous Agents and Multi-Agent Systems*, 17(1):1–10. ISSN 1573-7454. doi: 10.1007/s10458-008-9047-8. URL http://dx.doi.org/10.1007/s10458-008-9047-8. Cited on page 1

Boissier et al. (2013a) Olivier Boissier, Rafael H Bordini, Jomi F Hübner, Alessandro Ricci and Andrea Santi. Multi-agent oriented programming with JaCaMo. *Science of Computer Programming*, 78(6):747–761. Cited on page 13, 14

Boissier et al. (2013b) Olivier Boissier, Rafael H. Bordini, Jomi F. Hübner, Alessandro Ricci and Andrea Santi. Multi-agent oriented programming with jacamo. *Science of Computer Programming*, 78(6):747 – 761. ISSN 0167-6423. doi: https://doi.org/10.1016/j.scico.2011.10.004. URL http://www.sciencedirect.com/science/article/pii/S016764231100181X. Special section: The Programming Languages track at the 26th ACM Symposium on Applied Computing (SAC 2011) & Special section on Agent-oriented Design Methods and Programming Techniques for Distributed Computing in Dynamic and Complex Environments. Cited on page xi, 15, 16

53
Boissier et al. (2016) Olivier Boissier, Jomi F Hübner and Alessandro Ricci. The JaCaMo framework. In Social coordination frameworks for social technical systems, pages 125–151. Springer, Cham. doi: 10.1007/978-3-319-33570-4_7. URL http://dx.doi.org/10.1007/978-3-319-33570-4_7. Cited on page 25

Bordini et al. (2007) Rafael H Bordini, Jomi Fred Hübner and Michael Wooldridge. Programming multi-agent systems in AgentSpeak using Jason, volume 8. John Wiley & Sons. Cited on page 14, 15, 25

Bratman (1987) Michael Bratman. Intention, plans, and practical reason, volume 10. Harvard University Press Cambridge, MA. Cited on page 14

Broersen et al. (2001) Jan Broersen, Mehdi Dastani, Joris Hulstijn, Zisheng Huang and Leendert van der Torre. The BOID architecture: Conflicts between beliefs, obligations, intentions and desires. In Proceedings of the 5th International Conference on Autonomous Agents, pages 9–16, New York, NY. ACM Press. doi: 10.1145/375735.375766. URL http://dx.doi.org/10.1145/375735.375766. Cited on page 1

Bruhin et al. (2010) Adrian Bruhin, Helga Fehr-Duda and Thomas Epper. Risk and rationality: Uncovering heterogeneity in probability distortion. Econometrica, 78(4):1375–1412. Cited on page 12

Busetta et al. (1999) Paolo Busetta, Ralph Rönquist, Andrew Hodgson and Andrew Lucas. Jack intelligent agents-components for intelligent agents in java. AgentLink News Letter, 2(1):2–5. Cited on page 14

Campos et al. (2013) Jordi Campos, Maite Lopez-Sanchez, Maria Salamó, Pedro Avila and Juan A. Rodríguez-Aguilar. Robust regulation adaptation in multi-agent systems. ACM Transactions on Autonomic and Adaptive Systems, 8(3):13:1–13:27. doi: 10.1145/2517328. URL http://dx.doi.org/10.1145/2517328. Cited on page 17

Cardoso and Oliveira (2009) Henrique Lopes Cardoso and Eugénio Oliveira. Adaptive deterrence sanctions in a normative framework. In Proceedings of the 2009 IEEE/WIC/ACM International Joint Conference on Web Intelligence and Intelligent Agent Technology, pages 36–43, Washington, D.C. IEEE Computer Society. doi: 10.1109/WI-IAT.2009.123. URL http://dx.doi.org/10.1109/WI-IAT.2009.123. Cited on page 6, 17

Castelfranchi (1995) Cristiano Castelfranchi. Cognitive and social action. Psychology Press. Cited on page 1

Castelfranchi (2000) Cristiano Castelfranchi. Engineering social order. In Engineering societies in the agents world, pages 1–18. Springer. Cited on page 1

Castelfranchi and Falcone (1998) Cristiano Castelfranchi and Rino Falcone. Principles of trust for mas: Cognitive anatomy, social importance, and quantification. In Multi Agent Systems, 1998. Proceedings. International Conference on, pages 72–79. IEEE. Cited on page 5, 6

Centeno et al. (2011) Roberto Centeno, Holger Billhardt and Ramón Hermoso. An adaptive sanctioning mechanism for open multi-agent systems regulated by norms. In Proceedings of the IEEE 23rd International Conference on Tools with Artificial Intelligence, pages 523–530, Washington, D.C. IEEE Computer Society. doi: 10.1109/ICTAI.2011.85. URL http://doi.ieeecomputersociety.org/10.1109/ICTAI.2011.85. Cited on page 6, 17

Centeno et al. (2013) Roberto Centeno, Holger Billhardt and Ramón Hermoso. Persuading agents to act in the right way: An incentive-based approach. EAAI, 26(1):198–210. doi: 10.1016/j.engappai.2012.10.001. URL http://dx.doi.org/10.1016/j.engappai.2012.10.001. Cited on page 17

Cohen and Levesque (1990) Philip R Cohen and Hector J Levesque. Intention is choice with commitment. Artificial intelligence, 42(2-3):213–261. Cited on page 14
REFERENCES

Conte and Castelfranchi(2006) Rosaria Conte and Cristiano Castelfranchi. The mental path of norms. Ratio Juris, 19(4):501–517. Cited on page 5

Conte and Paolucci(2002) Rosaria Conte and Mario Paolucci. Reputation in artificial societies: Social beliefs for social order, volume 6. Springer Science & Business Media. Cited on page 6

Conte et al.(1998) Rosaria Conte, Cristiano Castelfranchi and Frank Dignum. Autonomous norm acceptance. Springer. Cited on page 1

Conte et al.(2013) Rosaria Conte, Giulia Andrighetto and Marco Campenni. Minding norms: Mechanisms and dynamics of social order in agent societies. Oxford University Press. Cited on page 6

Criado(2018) Natalia Criado. Resource-bounded norm monitoring in multi-agent systems. Journal of Artificial Intelligence Research, 62:153–192. Cited on page 7, 17

Criado et al.(2013) Natalia Criado, Estefania Argente, Pablo Noriega and Vicente Botti. MaNEA: A distributed architecture for enforcing norms in open MAS. Engineering Applications of Artificial Intelligence, 26(1):76–95. doi: 10.1016/j.engappai.2012.08.007. URL http://dx.doi.org/10.1016/j.engappai.2012.08.007. Cited on page 7, 17

Daskalopulu et al.(2002) Aspassia Daskalopulu, Theo Dimitrakos and Tom Maibaum. Evidence-based electronic contract performance monitoring. Group Decision and Negotiation, 11(6):469–485. doi: 10.1023/A:1020691116541. URL http://dx.doi.org/10.1023/ACitedon page 17

Dastani(2014) Mehdi Dastani. A Survey of Multi-agent Programming Languages and Frameworks, pages 213–233. Springer Berlin Heidelberg, Berlin, Heidelberg. ISBN 978-3-642-54432-3. doi: 10.1007/978-3-642-54432-3_11. URL https://doi.org/10.1007/978-3-642-54432-3_11. Cited on page 13, 14, 15

Dastani(2008) Mehdi Dastani. 2APL: a practical agent programming language. Autonomous agents and multi-agent systems, 16(3):214–248. Cited on page 14

Dastani et al.(2009) Mehdi Dastani, Davide Grossi, John-Jules Ch Meyer and Nick Tinnemeier. Normative multi-agent programs and their logics. In Knowledge Representation for Agents and Multi-Agent Systems, pages 16–31. Springer. Cited on page 15

Dignum(2004) Virginia Dignum. A model for organizational interaction: Based on agents, founded in logic. Ph.D. Thesis, Utrecht University. Cited on page 1

Dong-Huynha et al.(2004) T Dong-Huynha, N Jennings and N Shadbolt. Fire: An integrated trust and reputation model for open multi-agent systems. In 16th European Conference on Artificial Intelligence, Valencia, Spain, pages 18–22. IOS Press. Cited on page 6

Esteva et al.(2000) Marc Esteva, Juan A. Rodríguez-Aguilar, Josep L. Arcos, Carles Sierra and Pere Garcia. Institutionalizing open multi-agent systems. In Proceedings of the 4th International Conference on Multi-Agent Systems, pages 381–382, Boston, MA. IEEE Computer Society. doi: 10.1109/ICMAS.2000.858485. URL http://dx.doi.org/10.1109/ICMAS.2000.858485. Cited on page 1

Esteva et al.(2002) Marc Esteva, David De La Cruz and Carles Sierra. ISLANDER: an electronic institutions editor. In Proceedings of the first international joint conference on Autonomous agents and multiagent systems: part 3, pages 1045–1052. ACM. Cited on page 15

Esteva et al.(2004) Marc Esteva, Bruno Rosell, Juan A Rodriguez-Aguilar and Josep Li Arcos. AMELI: An agent-based middleware for electronic institutions. In Proceedings of the Third International Joint Conference on Autonomous Agents and Multiagent Systems-Volume 1, pages 236–243. IEEE Computer Society. Cited on page 15
Fagundes et al. (2014) Moser Silva Fagundes, Sascha Ossowski and Felipe Meneguzzi. Imperfect norm enforcement in stochastic environments: an analysis of efficiency and cost tradeoffs. In Ibero-American Conference on Artificial Intelligence, pages 523–535. Springer. Cited on page 7, 17

Ferber (1999) Jacques Ferber. Multi-Agent Systems: An Introduction to Distributed Artificial Intelligence. Addison-Wesley Longman Publishing Co., Inc., Boston, MA, USA, 1st edition. ISBN 0201360489. Cited on page 5

Fornara and Colombetti (2008) Nicoletta Fornara and Marco Colombetti. Specifying and enforcing norms in artificial institutions. In International Workshop on Declarative Agent Languages and Technologies, pages 1–17. Springer. Cited on page 1

Gasparini et al. (2018) Luca Gasparini, Timothy J Norman and Martin J Kollingbaum. Severity-sensitive norm-governed multi-agent planning. Autonomous Agents and Multi-Agent Systems, 32 (1):26–58. Cited on page 7, 17

Gâteau et al. (2005) Benjamin Gâteau, Olivier Boissier, Djamel Khadraoui and Eric Dubois. MOISE Inst: An organizational model for specifying rights and duties of autonomous agents. In Marie P. Gleizes, Gal A. Kaminka, Ann Nowé, Sascha Ossowski, Karl Tuyls and Katja Verbeeck, editors, Proceedings of the 3rd European Workshop on Multi-Agent Systems, pages 484–485, Brussels. Koninklijke Vlaamse Academie van Belie voor Wetenschappen en Kunsten. Cited on page 1

Giardini et al. (2014) Francesca Giardini, Mario Paolucci, Daniel Villatoro and Rosaria Conte. Punishment and gossip: Sustaining cooperation in a public goods game. In B. Kamiński and G. Koloch, editors, Advances in Social Simulation, volume 229 of Advances in Intelligent Systems and Computing, pages 107–118, Berlin. Springer-Verlag. doi: 10.1007/978-3-642-39829-2_10. URL http://dx.doi.org/10.1007/978-3-642-39829-2_10. Cited on page 29, 34

Giardini et al. (2015) Francesca Giardini, Mario Paolucci, Diana Adamatti and Rosaria Conte. Group Size and Gossip Strategies: An ABM Tool for Investigating Reputation-Based Cooperation, pages 104–118. Springer International Publishing, Cham. ISBN 978-3-319-14627-0. doi: 10.1007/978-3-319-14627-0_8. URL https://doi.org/10.1007/978-3-319-14627-0_8. Cited on page 34, 35

Gonzalez and Wu (1999) Richard Gonzalez and George Wu. On the shape of the probability weighting function. Cognitive psychology, 38(1):129–166. Cited on page 12

Grossi (2007) Davide Grossi. Designing invisible handcuffs: Formal investigations in institutions and organizations for multi-agent systems. Ph.D. Thesis, Utrecht University. Cited on page 15

Grossi et al. (2006) Davide Grossi, Huib Aldewereld and Frank Dignum. Ubi lex, ibi poena: Designing norm enforcement in e-institutions. In International Workshop on Coordination, Organizations, Institutions, and Norms in Agent Systems, pages 101–114. Springer. Cited on page 1

Hauert (2006) Christoph Hauert. Cooperation, collectives formation and specialization. Advances in Complex Systems, 9(04):315–335. Cited on page 29

Hendrikx et al. (2015) Ferry Hendrikx, Kris Bubendorfer and Ryan Chard. Reputation systems: A survey and taxonomy. Journal of Parallel and Distributed Computing, 75:184–197. Cited on page 6

Henrich et al. (2006) Joseph Henrich, Richard McElreath, Abigail Barr, Jean Ensminger, Clark Barrett, Alexander Bolyanatz, Juan Camilo Cardenas, Michael Gurven, Edwins Gwako, Natalie Henrich, Carolyn Lesorogol, Frank Marlowe, David Tracer and John Ziker. Costly punishment across human societies. Science, 312(5781):1767–1770. ISSN 0036-8075. doi: 10.1126/science.1127333. URL https://science.sciencemag.org/content/312/5781/1767. Cited on page 32

Hewitt (1991) Carl Hewitt. Open information systems semantics for distributed artificial intelligence. Artificial intelligence, 47(1-3):79–106. Cited on page 5
References

Hindriks (2009) Koen V Hindriks. Programming rational agents in GOAL. In Multi-agent programming, pages 119–157. Springer. Cited on page 14

Hindriks et al. (1999) Koen V Hindriks, Frank S De Boer, Wiebe Van der Hoek and John-Jules Ch Meyer. Agent programming in 3APL. Autonomous Agents and Multi-Agent Systems, 2(4):357–401. Cited on page 14

Hollander and Wu (2011) Christopher D Hollander and Annie S Wu. The current state of normative agent-based systems. Journal of Artificial Societies and Social Simulation, 14(2):6. Cited on page 1

Hubner et al. (2007) Jomi F. Hubner, Jaime S. Sichman and Olivier Boissier. Developing organised multiagent systems using the moise+ model: Programming issues at the system and agent levels. Int. J. Agent-Oriented Softw. Eng., 1(3-4):370–395. ISSN 1746-1375. doi: 10.1504/IJAOSE.2007.016266. URL http://dx.doi.org/10.1504/IJAOSE.2007.016266. Cited on page 16

Hübner et al. (2007) Jomi F Hübner, Jaime Sichman and Olivier Boissier. Developing organised multiagent systems using the MOISE+ model: programming issues at the system and agent levels. International Journal of Agent-Oriented Software Engineering, 1(3-4):370–395. Cited on page 15

Hübner et al. (2010) Jomi F Hübner, Olivier Boissier, Rosine Kitio and Alessandro Ricci. Instrumenting multi-agent organisations with organisational artifacts and agents. Autonomous agents and multi-agent systems, 20(3):369–400. Cited on page 15, 16

Johnson et al. (2012) Matthew Johnson, Jeffrey M Bradshaw, Paul J Feltovich, Catholijn Jonker, Birna Van Riemsdijk and Maarten Sierhuis. Autonomy and interdependence in human-agent-robot teams. Intelligent Systems, IEEE, 27(2):43–51. Cited on page 1

Jones and Sergot (1993) Andrew J. I. Jones and Marek Sergot. On the characterisation of law and computer systems: The normative systems perspective. In Deontic Logic in Computer Science: Normative System Specification, pages 275–307. John Wiley & Sons. Cited on page 15

Kafalý et al. (2016) Özgür Kafalý, Nirav Ajmeri and Munindar P Singh. Revani: . IEEE Intelligent Systems, 31(5):8–15. Cited on page 1

Kahneman and Tversky (1979) Daniel Kahneman and Amos Tversky. Prospect theory: An analysis of decision under risk. Econometrica, 47(2):263–91. Cited on page 10, 11

Kollingbaum and Norman (2003) Martin J. Kollingbaum and Timothy J. Norman. NoA - a normative agent architecture. In Georg Gottlob and Toby Walsh, editors, Proceedings of the 18th International Conference on Artificial Intelligence, pages 1465–1466, San Francisco, CA. Morgan Kaufmann. Cited on page 1

Kőszegi and Rabin (2006) Botond Kőszegi and Matthew Rabin. A model of reference-dependent preferences. The Quarterly Journal of Economics, 121(4):1133–1165. Cited on page 12, 13, 30

Kőszegi and Rabin (2007) Botond Kőszegi and Matthew Rabin. Reference-dependent risk attitudes. American Economic Review, 97(4):1047–1073. Cited on page 12, 30

Kőszegi and Rabin (2009) Botond Kőszegi and Matthew Rabin. Reference-dependent consumption plans. American Economic Review, 99(3):909–36. Cited on page 12, 13, 30

Ledyard (1995) John Ledyard. Public goods: A survey of experimental research. In John H. Kagel and Alvin E. Roth, editors, The Handbook of Experimental Economics, pages 111–194. Princeton University Press, Princeton, NJ. Cited on page 3, 29
REFERENCES

López and Luck(2003) Fabiola L. y López and Michael Luck. Modelling norms for autonomous agents. In Edgar Chávez, Jesús Favela, Marcelo Mejia and Alberto Oliart, editors, Proceedings of the 4th Mexican International Conference on Computer Science, pages 238–245, Washington, D.C. IEEE Computer Society. doi: 10.1109/ENC.2003.1232900. URL http://dx.doi.org/10.1109/ENC.2003.1232900. Cited on page 7, 17

Mahmoud et al.(2012a) Samhar Mahmoud, Nathan Griffiths, Jeroen Keppens and Michael Luck. Efficient norm emergence through experiential dynamic punishment. In Luc De Raedt, Christian BessiÈre, Didier Dubois, Patrick Doherty, Paolo Frasconi, Fredrik Heintz and Peter J. F. Lucas, editors, Proceedings of the 20th European Conference on Artificial Intelligence - Including Prestigious Applications of Artificial Intelligence System Demonstrations Track, volume 242 of Frontiers in Artificial Intelligence and Applications, pages 576–581, Monpellier. IOS Press. doi: 10.3233/978-1-61499-098-7-576. URL http://dx.doi.org/10.3233/978-1-61499-098-7-576. Cited on page 7, 17

Mahmoud et al.(2012b) Samhar Mahmoud, Daniel Villatoro, Jeroen Keppens and Michael Luck. Optimised reputation-based adaptive punishment for limited observability. In Proceedings of the 6th IEEE International Conference on Self-Adaptive and Self-Organizing Systems, pages 129–138, Washington, DC. IEEE Computer Society. doi: 10.1109/SASO.2012.24. URL http://dx.doi.org/10.1109/SASO.2012.24. Cited on page 7, 17

Mann and Whitney(1947) Henry B Mann and Donald R Whitney. On a test of whether one of two random variables is stochastically larger than the other. The annals of mathematical statistics, pages 50–60. Cited on page 40

Meneguzzi et al.(2015) Felipe Meneguzzi, Odinaldo Rodrigues, Nir Oren, Wamberto W Vasconcelos and Michael Luck. BDI reasoning with normative considerations. Engineering Applications of Artificial Intelligence, 43:127–146. Cited on page 14

Meyer and Wieringa(1993) J.J.C. Meyer and R. Wieringa. Deontic logic in computer science: normative system specification. Wiley professional computing. J. Wiley. ISBN 9780471937432. URL https://books.google.com.br/books?id=0YRQAAAAMAAJ. Cited on page 5

Modgil et al.(2009) Sanjay Modgil, Noura Faci, Felipe Meneguzzi, Nir Oren, Simon Miles and Michael Luck. A framework for monitoring agent-based normative systems. In Proceedings of the 8th International Conference on Autonomous Agents and Multiagent Systems, pages 153–160, Richland, SC. International Foundation for Autonomous Agents and Multiagent Systems. Cited on page 6, 17

Modgil et al.(2015) Sanjay Modgil, Nir Oren, Noura Faci, Felipe Meneguzzi, Simon Miles and Michael Luck. Monitoring compliance with e-contracts and norms. Artificial Intelligence and Law, 23(2):161–196. Cited on page 6, 17

Morewedge and Giblin(2015) Carey K Morewedge and Colleen E Giblin. Explanations of the endowment effect: an integrative review. Trends in cognitive sciences, 19(6):339–348. Cited on page 13

Morgenstern and Von Neumann(1953) Oskar Morgenstern and John Von Neumann. Theory of games and economic behavior. Princeton university press. Cited on page 11

Mui et al.(2003) Lik Mui, Ari Halberstadt and Mojdeh Mohtashemi. Evaluating reputation in multi-agents systems. In Proceedings of the 2002 International Conference on Trust, Reputation, and Security: Theories and Practice, AAMAS’02, pages 123–137, Berlin, Heidelberg. Springer-Verlag. ISBN 3-540-00988-4. URL http://dl.acm.org/citation.cfm?id=1762128.1762138. Cited on page 6
Nair and Tambe(2005) Ranjit Nair and Milind Tambe. Hybrid BDI-POMDP framework for multiagent teaming. *Journal of Artificial Intelligence Research*, 23:367–420. Cited on page 14

Nardin et al.(2016) Luis G. Nardin, Tina Balke, Nirav Ajmeri, Anup A. Kalia, Jaime S. Sichman and Munindar P. Singh. Classifying sanctions and designing a conceptual sanctioning process model for socio-technical systems. *The Knowledge Engineering Review*, 31(2):142–166. doi: 10.1017/S0269888916000023. URL http://dx.doi.org/10.1017/S0269888916000023. Cited on page iii, v, xi, 1, 2, 3, 7, 9, 19, 23, 26, 45

Nardin(2015) Luis Gustavo Nardin. *An Adaptive Santioning Enforcement Model for Normative Multiagent Systems*. Ph.D. Thesis, Universidade de São Paulo. Cited on page xiii, 2, 6, 16, 17, 26

Noriega(1997) Pablo Noriega. *Agent mediated auctions: The fishmarket metaphor*. Ph.D. Thesis, Universitat Autònoma de Barcelona. Cited on page 1

Omicini et al.(2008) Andrea Omicini, Alessandro Ricci and Mirko Viroli. Artifacts in the a&ca meta-model for multi-agent systems. *Autonomous agents and multi-agent systems*, 17(3):432–456. doi: 10.1007/11423355_10. URL http://dx.doi.org/10.1007/11423355_10. Cited on page 15

Parunak and Weyns(2007) H Van Dyke Parunak and Danny Weyns. Guest editors’ introduction, special issue on environments for multi-agent systems. *Autonomous Agents and Multi-Agent Systems*, 14(1):1–4. Cited on page 15

Pasquier et al.(2005) Philippe Pasquier, Roberto A. Flores and Brahim Chaib-draa. Modelling flexible social commitments and their enforcement. In Marie P. Glezies, Andrea Omicini and Franco Zambonelli, editors, *Proceedings of the 5th international conference on Engineering Societies in the Agents World (ESAW’04)*, volume 3451 of *Lecture Notes in Computer Science*, pages 139–151, Berlin, DE. Springer-Verlag. doi: 10.1007/11423355_10. URL http://dx.doi.org/10.1007/11423355_10. Cited on page 2

Pasquier et al.(2006) Philippe Pasquier, Roberto A Flores and Brahim Chaib-draa. An ontology of social control tools. In *Proceedings of the fifth international joint conference on Autonomous agents and multi-agent systems*, pages 1369–1371. ACM. Cited on page 5

Pinninck et al.(2010) Adrián P. de Pinninck, Carles Sierra and Marco Schorlemmer. A multiagent network for peer norm enforcement. *JAAMAS*, 21(3):397–424. doi: 10.1007/s10458-009-9107-8. URL http://dx.doi.org/10.1007/s10458-009-9107-8. Cited on page 17

Pinyol and Sabater-Mir(2013) Isaac Pinyol and Jordi Sabater-Mir. Computational trust and reputation models for open multi-agent systems: a review. *Artificial Intelligence Review*, 40(1):1–25. Cited on page 6

Pinyol et al.(2012) Isaac Pinyol, Jordi Sabater-Mir, Pilar Dellunde and Mario Paolucci. Reputation-based decisions for logic-based cognitive agents. *Autonomous Agents and Multi-Agent Systems*, 24(1):175–216. Cited on page 6

Rabin(2013) Mathew Rabin. Risk aversion and expected-utility theory: A calibration theorem. In *Handbook of the Fundamentals of Financial Decision Making: Part I*, pages 241–252. World Scientific. Cited on page 11

Rao(1996) Anand S Rao. Agentspeak (I): Bdi agents speak out in a logical computable language. In *European Workshop on Modelling Autonomous Agents in a Multi-Agent World*, pages 42–55. Springer. Cited on page 15

Rao and Georgeff(1991) Anand S Rao and Michael P Georgeff. Modeling rational agents within a BDI-architecture. *KR*, 91:473–484. Cited on page 14
REFERENCES

Ricci et al. (2009) Alessandro Ricci, Michele Piunti, Mirko Viroli and Andrea Omicini. Environment programming in CArtAgO. In Multi-Agent Programming, pages 259–288. Springer. Cited on page 15, 25

Ricci et al. (2011) Alessandro Ricci, Michele Piunti and Mirko Viroli. Environment programming in multi-agent systems: an artifact-based perspective. Autonomous Agents and Multi-Agent Systems, 23(2):158–192. ISSN 1573-7454. doi: 10.1007/s10458-010-9140-7. URL https://doi.org/10.1007/s10458-010-9140-7. Cited on page 15

Sabater and Sierra (2001) Jordi Sabater and Carles Sierra. Social regret, a reputation model based on social relations. ACM SIGecom Exchanges, 3(1):44–56. Cited on page 6

Sabater et al. (2006) Jordi Sabater, Mario Paolucci and Rosaria Conte. Repage: Reputation and image among limited autonomous partners. Journal of artificial societies and social simulation, 9(2). Cited on page 6

Shafir and LeBeouf (2002) Eldar Shafir and Robyn A LeBoeuf. Rationality. Annual review of psychology, 53(1):491–517. Cited on page 10

Shoham (1993) Yoav Shoham. Agent-oriented programming. Artificial intelligence, 60(1):51–92. Cited on page 14

Shoham and Tennenholz (1992) Yoav Shoham and Moshe Tennenholz. On the synthesis of useful social laws for artificial agent societies (preliminary report). In AAAI, pages 276–281. Cited on page 5

Szabó and Hauert (2002) György Szabó and Christoph Hauert. Phase transitions and volunteering in spatial public goods games. Physical review letters, 89(11):118101. Cited on page 29

Tasaki et al. (2010) Makoto Tasaki, Yuichi Yabu, Yuki Iwanari, Makoto Yokoo, Janusz Marecki, Pradeep Varakantham and Milind Tambe. Introducing communication in dis-pomdps with locality of interaction. Web Intelligence and Agent Systems: An International Journal, 8(3):303–311. Cited on page 14

Tversky and Kahneman (1992) Amos Tversky and Daniel Kahneman. Advances in prospect theory: Cumulative representation of uncertainty. Journal of Risk and uncertainty, 5(4):297–323. Cited on page xi, 10, 11, 12, 13

Vercouter and Muller (2010) Laurent Vercouter and Guillaume Muller. Liar: achieving social control in open and decentralized multiagent systems. Applied Artificial Intelligence, 24(8):723–768. Cited on page 6

Verhagen (2000) Harko Verhagen. Norm autonomous agents. Ph.D. Thesis, Citeseer. Cited on page 1, 5

Villatoro et al. (2011) Daniel Villatoro, Giulia Andrighetto, Jordi Sabater-Mir and Rosaria Conte. Dynamic sanctioning for robust and cost-efficient norm compliance. In Proceedings of the 22th International Joint Conference on Artificial Intelligence, pages 414–419, Menlo Park, CA. AAAI Press. doi: 10.5591/978-1-57735-516-8/IJCAI11-077. URL http://dx.doi.org/10.5591/978-1-57735-516-8/IJCAI11-077. Cited on page 7, 17

Wang and Singh (2010) Yonghong Wang and Munindar P Singh. Evidence-based trust: A mathematical model geared for multiagent systems. ACM Transactions on Autonomous and Adaptive Systems (TAAS), 5(4):14. Cited on page 6

Weyns et al. (2004) Danny Weyns, H Van Dyke Parunak, Fabien Michel, Tom Holvoet and Jacques Ferber. Environments for multiagent systems state-of-the-art and research challenges. E4MAS, 3374:1–47. Cited on page 15
Weyns et al. (2007) Danny Weyns, Andrea Omicini and James Odell. Environment as a first class abstraction in multiagent systems. *Autonomous Agents and Multi-Agent Systems*, 14(1):5–30. ISSN 1573-7454. doi: 10.1007/s10458-006-0012-0. URL https://doi.org/10.1007/s10458-006-0012-0. Cited on page 15

Winikoff (2005) Michael Winikoff. JACK™ intelligent agents: an industrial strength platform. In *Multi-Agent Programming*, pages 175–193. Springer. Cited on page 14

Wooldridge (2009) Michael Wooldridge. *An introduction to multiagent systems*. John Wiley & Sons. Cited on page 1

Wright (1963) G.H. Wright. *Norm and action: a logical enquiry*. International library of philosophy and scientific method. Humanities. URL https://books.google.com.br/books?id=6VMYAAAAIAAJ. Cited on page 5

Zacharia and Maes (2000) Giorgos Zacharia and Pattie Maes. Trust management through reputation mechanisms. *Applied Artificial Intelligence*, 14(9):881–907. Cited on page 6