

A Framework for Automatic Monitoring of Norms and Agreements Concerning Digital Assets in the Web of Data

A dissertation presented by
Soheil Roshankish

Supervised by
Dr. Nicoletta Fornara

Co-Supervised by
Prof. Marco Colombetti
Prof. Lorenzo Cantoni

Submitted to the
Faculty of Communication, Culture and Society
Università della Svizzera italiana

for the degree of
Ph.D. in Communication Sciences
specialization in “ICT – Information and Communication Technologies”

September 2024

Thesis committee

Prof. Andrea Rocci, Università della Svizzera italiana, Switzerland (President)

Prof. Víctor Rodríguez Doncel, Universidad Politécnica de Madrid (External Reviewer)

Prof. Matteo Baldoni, Università di Torino (External Reviewer)

Dr. Nicoletta Fornara, Università della Svizzera italiana, Switzerland (Thesis Director)

Prof. Marco Colombetti, Università della Svizzera italiana (Co-Director)

Prof. Lorenzo Cantoni, Università della Svizzera italiana (Co-Director)

Acknowledgements

I begin this Ph.D. journey with a profound sense of gratitude and appreciation for the many individuals who have played pivotal roles in shaping this research and its achievements. First and foremost, I express my deepest thanks to my supervisor, Dr. Nicoletta Fornara, whose guidance, mentorship, and insightful feedback have been pivotal in shaping the foundation of this thesis. Their unwavering support and encouragement have been a constant source of inspiration.

I am equally grateful for the mentorship provided by my two prior thesis supervisors, Prof. Marco Colombetti and Prof. Lorenzo Cantoni. Their guidance and expertise have played an instrumental role in my academic development, laying the groundwork for the research skills and methodologies that I bring to this Ph.D. thesis.

I extend my appreciation to the faculty members of Communication, Culture, and Society for their valuable insights and contributions, shaping my academic perspective and enriching the scholarly discourse. The collaborative spirit of my colleagues at the Institute of Digital Technologies for Communication has fostered an environment of intellectual exchange, and I am grateful for the shared journey of knowledge exploration.

Furthermore, I am grateful to the Swiss National Foundation for their financial support, which has allowed me to dedicate myself fully to this research.

A heartfelt acknowledgment is extended to my parents, Giti and Asad, for their support, encouragement, and sacrifices that have been foundational to my academic journey. Their belief in my abilities and the values they instilled in me form the core of my academic pursuits.

I am also grateful to my sisters, Gita and Sanaz, along with their husbands, Omid and Kianoush, for their support and encouragement. Their belief in my capabilities has been a driving force, serving as a constant motivator throughout my academic journey.

My gratitude extends to all family and friends for their understanding, encouragement, and unwavering support, making this journey more meaningful. To my cousin Ladan, thank you for the positive energies you have sent me all these years.

Special appreciation is expressed to my fiancée, Maryam, for her unwavering love, understanding, and encouragement. Her belief in my dreams and patience during challenging times have made this academic pursuit even more meaningful. I am truly fortunate to have her by my side.

I also want to express my profound gratitude to my late uncles, Hosein and Hooshang, for their invaluable support and guidance during my educational journey, leaving an enduring impact on my pursuit of knowledge.

Additionally, I extend sincere appreciation to Uncle Saied, for his continuous support and unwavering belief in my academic ventures. His encouragement and wisdom have been a source of inspiration, providing me with the strength to overcome challenges and strive for excellence.

As I embark on this Ph.D. thesis, I am humbled by the collective efforts and contributions of those around me. This thesis is not only a product of individual dedication but also a reflection of the collaborative and supportive academic community I am fortunate to be a part of.

Thank you all for being an integral part of the beginning of this academic journey.

Abstract

In today's digital economy, a significant portion of human activities relies on digital devices and assets. However, it is insufficient to make digital assets public without specifying constraints on their usage and access. The problem is that lacking a license or agreement does not grant the unrestricted right to manipulate such data. While licenses and agreements can be expressed in human-readable formats, the growing exchange of digital assets necessitates more formal and machine-readable mechanisms. These mechanisms facilitate machine-to-machine interactions and enable valuable services such as advanced resource searches based on licenses, aggregation of resources under different licenses, and automatic verification of normative or legal relations within the chain of interactions among data producers, publishers, and consumers.

To provide these services effectively, it is crucial not only to propose the syntax of a formal language for expressing licenses, norms, and agreements but also to establish formal semantics for automated reasoning on these constraints. In this thesis, we mainly propose a new model of norms called T-Norm (where T stands for temporal) which offers a flexible and robust approach to regulating actions with temporal constraints, allowing norm designers to formalize their norms using semantic web technologies to enhance the model's expressiveness and reasoning capabilities.

This thesis provides a comprehensive overview of the cumulative thesis, consisting of four papers that contribute to the fields of Normative Multi-agent Systems and Human-Computer Interaction. Three papers are contributions to the Normative Multi-agent Systems Community. Chapter 2 proposes the T-NORM model, a formalization of relational norms with temporal constraints. This model, published in the Proceedings of the 25th International Workshop on Coordination, Organizations, Institutions, Norms and Ethics for Governance of Multi-Agent Systems (COINE) in 2021, introduces temporal constraints, uses Semantic Web Technologies, and offers operational semantics, making norms machine-readable.

Chapter 3 continues the work on the T-Norm model by presenting a methodology for formalizing norms in Multi-agent Systems. This work was published in 19th European

Conference on Multi-agent Systems (EUMAS 2022), it addresses the gap in facilitating the usage of the T-Norm model.

Chapter 4 builds on previous work, proposing a rich set of norm types that could be used to study the expressive power of different formal models of norms and to compare them, and refine the methodology with sections devoted to the induction of normative power and the formalization of exceptions and permissions. This paper has been published in the journal SN Computer Science, published by Springer Nature

Chapter 5 shifts focus to digital fashion, discussing privacy concerns in the increasing use of digital technologies in the fashion field. It highlights challenges in customer privacy and proposes the use of Semantic Web Technologies and W3C Open Digital Rights Language (ODRL) for transforming privacy policies into machine-readable structures. The chapter also stresses the importance of policy monitoring and outlines components for automatic monitoring.

Chapter 6 concludes the thesis and proposes possible future work perspectives that could impact positively to the future of W3C Open Normative Multi-agent Systems (NorMAS) community. Overall, the thesis contributes significantly to Normative Multi-agent Systems and Human-Computer Interaction, providing valuable insights, methodologies, and models for formalizing norms and addressing privacy concerns in the digital fashion domain.

Keywords: Artificial Intelligence, Normative Multi-agent Systems, Norms, Policies, Rules, Human-Computer Interaction, Semantic Web Technologies

Table of Contents

1	Introduction	1
1.1	Goals of the thesis	1
1.2	Research Questions	2
1.3	Thesis structural overview	4
1.4	Background	10
1.4.1	Multi-agent Systems	10
1.4.2	Norms and Policies in Multi-agent Systems: An introduction . . .	10
1.4.3	Normative Multi-agent Systems	15
1.5	Semantic Web Technologies and Multi-agentsystems	17
1.5.1	What are Semantic Web Technologies?	18
1.5.2	Impact of Semantic Web Technologies on Multi-agent Systems .	21
1.6	Current Problems in Normative Systems	23
1.7	Related works	26
2	A Framework for Automatic Monitoring of Norms that Regulate Time Constrained Actions	37
2.1	Abstract	38
2.2	Introduction	38
2.3	Design Goals	41
2.4	The T-Norm model of norms	42
2.4.1	Expressive power of the model	47
2.4.2	The model of Exceptions	50
2.5	Operational Semantics of the model	53
2.6	Architecture of the framework and its implementation	56
2.7	Related Work	59

3	A Methodology for Formalizing Different Types of Norms	63
3.1	Abstract	64
3.2	Introduction	64
3.3	Related works	66
3.4	The T-Norm model	68
3.5	Methodology	70
3.5.1	Using ontologies for modeling norms	71
3.5.2	Norms with activation condition	74
3.5.3	Representing obligations and prohibitions	77
3.5.4	Temporal aspect of norms	78
3.5.5	Specific and General Deontic Relations	80
3.6	Conclusions	81
4	How to Formalize Different Types of Norms in Multi-Agent Systems: a Methodology Focused on the T-Norm Model	83
4.1	Abstract	85
4.2	Introduction	85
4.3	Related works	90
4.4	Running examples	95
4.5	Types of Norms	95
4.5.1	Norms activated by an event or state of affairs or both	96
4.5.2	Norms defining specific or general obligation or prohibition	97
4.5.3	Norms expressing a temporal constraint for regulated actions	99
4.5.4	Norms that induce normative powers	100
4.6	The T-Norm model	102
4.7	Methodology	107
4.7.1	Using ontologies for modeling norms	107
4.7.2	Formalizing the activation condition	111
4.7.3	Representing obligations and prohibitions	114
4.7.4	Representing the temporal constraints of norms	116

4.7.5	Formalizing specific or general obligation or prohibition	119
4.7.6	Formalizing norms inducing normative power	120
4.7.7	Representing permissions and exemptions	122
4.8	Conclusions	125
5	Exploration of Norms and Policies in Digital Fashion Domain Using Semantic Web Technologies	129
5.1	Abstract	131
5.2	Introduction	131
5.3	Methodology	133
5.4	Semantic Web Technologies in Digital Fashion	134
5.4.1	Semantic Web and Semantic Web Technologies	135
5.4.2	Related Works Using Semantic Web Technologies in Digital Fashion	137
5.5	The Open Digital Rights Language	138
5.5.1	The ODRL Information Model	138
5.5.2	Examples of ODRL policies for the fashion domain	139
5.6	Monitoring Norms and Policies	141
5.7	Conclusion	143
6	Conclusion	145
6.1	Research Contribution	146
6.2	Research Limitations	150
6.3	Future Work	152
	References	154

List of Figures

1.1	Academic contributions of this thesis (Green papers contributed to the NorMAS community and the yellow one to Human-Computer Interaction Community)	5
2.1	Architecture of the framework designed to compute the fulfillment or violation of norms.	57
2.2	Fulfillment/violation of the different deontic relations created by the activations of Norm01.	59
3.1	The T-NORM Ontology and its connections with other ontologies.	71
4.1	The T-NORM Ontology and its connections with other ontologies.	105
5.1	An RDF graph for describing customer data	136
5.2	Zalando data collection policy when a customer contacts them	140
5.3	Representation of the permission to archive customer's contact details such as email and phone number in ODRL	140
5.4	Representation of 5.2 in ODRL for customer David and CompanyX for the use of David's hometown information	141
5.5	Formalization of one permission and one prohibition in a policy agreement between CompanyX and Customer David.	142

1

Introduction

1.1 Goals of the thesis

The research topic of this thesis is deeply interdisciplinary and concerns the study and the formalization of application-independent models of norms that can be used to realize open Multi-agent Systems/platforms or agent societies, i.e. systems composed of multiple interacting computing elements, known as autonomous agents [163].

Those interacting agents may be software agents (such as personal assistants) able to decide for themselves what to do in order to represent the interests of their owner or they may be human agents. This field of research on Multi-agent Systems is interdisciplinary because the studies on concepts like norms, contracts, agreements, policies, organizations, institutions, and communication languages can be found in various research fields such as artificial intelligence, communication sciences, digital law, philosophy, and economics.

Such a field of research is relevant for our societies because thanks to the advent of the Internet, numerous social media applications, e-commerce systems, and platforms can be modeled as Multi-agent Systems where agents can arrive, get information on the rule of interactions, and on the policies or law that regulate the exchange of digital resources, and interact with other agents by deciding whether or not to abide by those rules.

Inspired by human society, an important aspect in the design and development of open Multi-agent Systems is the explicit (or declarative) formalization of the norms, rules, contracts, licenses, policies, and agreements governing the interaction between autonomous parties, together with the capacity to perform automatic reasoning on the semantics of

1.2. Research Questions

those concepts. These studies are relevant for a field of research called Normative Multi-agent Systems (NorMAS), a subfield of Multi-agent Systems. In NorMAS, automatic reasoning is important for being able to provide useful services on a set of norms, such as checking that the behavior of the agents in a system complies with its rules (for example checking that the receiver of a set of market data complies with the permissions attached to the set) or allowing an agent to plan its actions taking into account the interaction rules of the various systems in which it is involved.

Starting from the literature in the field and initially considering a set of types of examples of norms from our everyday life, the first goal of this thesis has been to propose an application-independent model of norms capable of overcoming the limits of expressivity of existing models and to propose an operational semantics for such a model in order to make it possible to realize a framework capable of monitoring the fulfillment and violations of the deontic relations created among the various agents during their interactions.

The second objective has been to propose a methodology for using the proposed model and other existing complementary models in the non-trivial task of formalizing existing norms (such as articles of law or licenses and rules governing the use and exchange of digital assets). This goal is closely linked to the third one, namely to be able to define a list of existing types of different norms in order to be able to understand which models can be used for the formalization of a certain norm and to think about the possibility of translating the formalization of a norm by using one model into the formalization of the same norm in another model. Finally, to test the feasibility of using norms and policy models in real applications, we investigated the problem of formalizing privacy policies in the field of fashion by using an existing model of norms.

1.2 Research Questions

To reach the goals of this thesis, Normative Multi-agent Systems should be examined through the following research questions. Some potential research questions that have be

1.2. Research Questions

explored within the interdisciplinary field of Normative Multi-agent Systems and the formalization of norms for open Multi-agent Systems could be categorized as:

- **Model Development:**

***RQ1.** What are the limitations of existing models in expressing and formalizing norms in Multi-agent Systems?*

***RQ2.** How can an application-independent model of norms be designed to address the expressivity limitations of current models?*

This research question examines the shortcomings of current normative models within Multi-agent Systems. By identifying these limitations, the study aims to propose an application-independent normative model that not only addresses these shortcomings but also provides an operational semantics for effective monitoring of deontic relations.

- **Methodology and Formalization:**

***RQ3.** What methodology can be devised for effectively using the proposed normative model and integrating it with complementary models to formalize existing norms?*

***RQ4.** What challenges arise in the non-trivial task of formalizing diverse norms, and how can these challenges be addressed?*

Based on the proposed normative model, this research investigates the practical application of norms by developing a methodology for their formalization. It explores how existing norms can be systematically formalized using this model, enhancing the understanding and implementation of normative frameworks. Moreover, the study seeks to bridge the gap between theoretical models and real-world normative practices in Multi-agent Systems.

- **Types of Norms and Model representation:**

***RQ5.** Can a comprehensive list of various types of norms be created to understand which models are suitable for their formalization?*

***RQ6.** How feasible is it to translate the formalization of a norm from one model to another, and under what conditions is such translation possible?*

This research question aims to categorize and understand various types of norms, providing a comprehensive list that can guide researchers in selecting suitable models for formalization. Additionally, the study explores the feasibility of translating the formalization of norms from one model to another. This comprehensive method represents how normative systems work and how they can be adapted or modified to fit various models or contexts.

- **Real-World Applications:**

***RQ7.** How can norms and policy models be practically applied in real-world scenarios, with a focus on privacy policies?*

This research question is specifically related to our contribution to the field of fashion (Chapter 5). Focusing on the practical implementation of norms, particularly in privacy policies within the fashion industry, this research question investigates challenges and solutions. By examining the formalization of privacy policies, the study aims to provide insights into the application of normative models in real-world scenarios, contributing to the development of privacy-aware Multi-agent Systems.

In this thesis, our goal is to answer the research questions outlined in this chapter in the coming chapters. As previously mentioned, given the interdisciplinary nature of this thesis, we believe that enhanced cooperation among diverse scientific communities is necessary to uncover the most effective answers. In the *Conclusion* section of this thesis, we will discuss how we addressed these questions by our contributions.

1.3 Thesis structural overview

This section provides an overview of the thesis chapters and their content. Four papers will be presented in this cumulative thesis. Three of them were published in proceeding

1.3. Thesis structural overview

of international workshops and conferences and one of them has been published in the peer-reviewed journal SN Computer Science by Springer Nature.

As it is represented in Figure 1.1 the first three papers are dedicated to the Normative Multi-agent Systems Community (Chapters 2-4). In the fourth paper, we present our contribution to the Human-Computer Interaction Community and more specifically address researchers interested in the theory and practice of autonomous agents and Multi-agent Systems, emphasizing formalization and reasoning about norms and policies in the fashion domain.

Chapter 1 outlines the objectives of this thesis and poses the key research questions. It then introduces the primary areas of focus: Multi-agent Systems, Normative Multi-agent Systems, and Semantic Web Technologies, with particular emphasis on their relevance to normative systems. Additionally, this chapter will review related works in the field of Normative Multi-agent Systems.

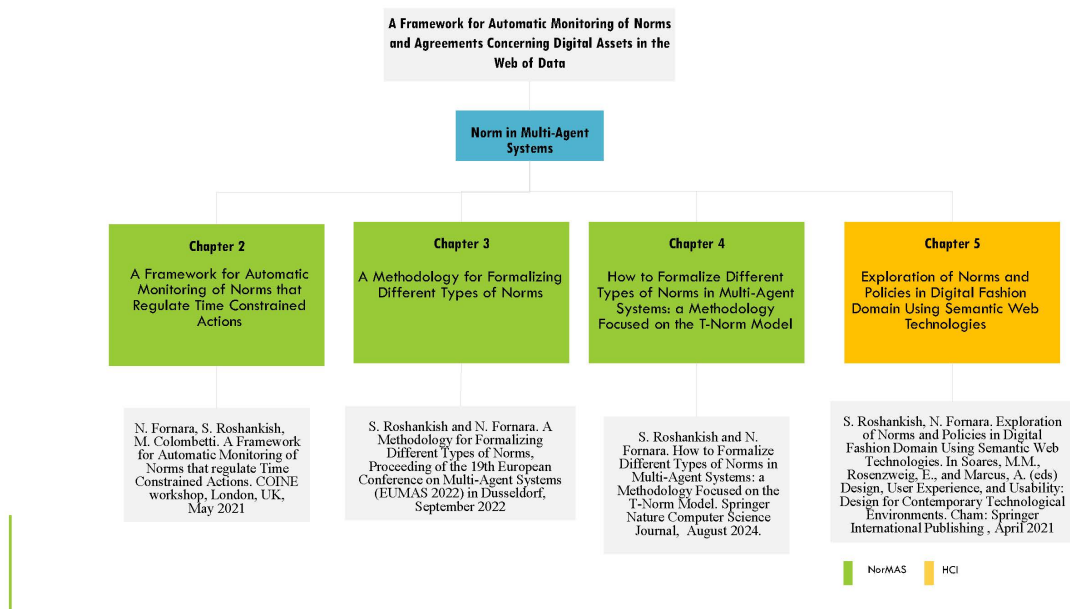


Fig. 1.1: Academic contributions of this thesis (Green papers contributed to the NorMAS community and the yellow one to Human-Computer Interaction Community)

1.3. Thesis structural overview

In Chapter 2 of this thesis, the main focus is on proposing a model of norms and a framework for automatically computing if any violation or fulfillment occurs between agents' interactions. As our first goal, this model has to express abstract ¹ norms and be able to regulate classes of actions that should or should not be performed. The second goal of this research which highlights this work in our research community is to define a model of norms able to regulate classes of actions whose performance is temporally constrained.

This model named the T-NORM model, where "T" stands for Temporal, is a formalization of relational norms that regulate classes of actions performed by agents in a society. These norms are related to actions such as paying, lending, entering restricted areas, etc. The key features of the proposed model include:

- **Temporal Constraints:** The model is designed to regulate actions within specific temporal intervals, a distinguishing feature. This means that it can monitor actions that are time-constrained, such as payment deadlines in e-commerce;
- **Use of Semantic Web Technologies:** The model uses semantic web languages, specifically OWL (Web Ontology Language), to formalize norms. This allows for automatic reasoning and together with the help of production rules they provide formal semantics for monitoring norms;
- **Expressiveness:** Instead of labeling norms as obligations or prohibitions, the model allows norm designers to express the sequence of events leading to fulfillment or violation. This flexibility enables a wide range of normative relations to be expressed;
- **Operational Semantics:** The article defines the operational semantics of the T-NORM model, combining OWL reasoning and forward chaining. This enables the automatic monitoring of the norms.
- **Machine-Readable Norms:** Formalizing norms into a machine-readable structure, enables automated analysis and querying. This is particularly important for areas

¹In this thesis using the term *abstract* such as "abstract norm" or "abstract model", implies a level of generalization and detachment from specific instances, allowing for broader applicability and theoretical exploration

1.3. Thesis structural overview

like privacy policies, where monitoring compliance is crucial, especially in the context of sensitive data (such as customer's personal information);

The main contributions of our first article (Chapter 2) are the introduction of the T-Norm model itself and its operational semantics. It offers a flexible and robust approach to regulating actions with temporal constraints, allowing norm designers to specify rules for fulfilling or violating norms. Additionally, the use of semantic web technologies enhances the model's expressiveness and reasoning capabilities. This research was published in the Proceedings of the 25th International Workshop on Coordination, Organizations, Institutions, Norms and Ethics for Governance of Multi-Agent Systems (COINE) in 2021.

The T-Norm model has been used for the formalization of a set of norms and a framework for the simulation of the evolution in time of those norms based on the actions performed by the agents and a set of events has been implemented and tested in [63]. Having the help of a straightforward translation process, the implemented system offers the ability to monitor the violations or fulfillments of rules generated using the T-Norm model and transformed into production rules. They suggested an architecture through combining OWL reasoning with a forward chaining interpreter of production rules for the first time which brings innovation to the field of normative multi-agent systems literature.

After the introduction of the T-Norm model to our research community and being an active member of W3C ODRL community group (where ODRL stands for Open Digital Rights Language), these helped us contact data scientists in the industry as well. The W3C ODRL Community Group is a part of the World Wide Web Consortium (W3C) that focuses on the development and standardization of the Open Digital Rights Language. ODRL is a framework for expressing rights and permissions related to digital content and services in a machine-readable format. The community group works on specifications, guidelines, and best practices to support the interoperability of digital rights management (DRM) systems.² As a result, we could receive positive feedback from both academic researchers and data scientists. However, based on the feedback, we felt a gap for a methodology that

²<https://www.w3.org/community/odrl/> community group

could help norm designers ³ to ease the usage of the T-Norm model in their research or work.

In Chapter 3, we filled this gap with our second publication which was accepted to be presented and published in the proceedings of the 19th European Conference on Multi-agent Systems (EUMAS 2022). This paper underscores the importance of formalizing norms in Multi-agent Systems and introduces a methodology for this purpose. It also emphasizes the benefits of machine-readable norms and the need for models of norms to be expressive enough to cover various types of deontic relations. This methodology involves several key aspects:

- **Using Ontologies for Modeling Norms:** The first step in formalizing norms is to define or select formal ontologies that represent the classes of events or actions relevant to the norm. These ontologies are used to specify the activation conditions, actions regulated by the norm, and constraints on actions. The W3C Web Ontology Language (OWL) is suggested as a suitable language for this purpose;
- **Norms with Activation Condition:** It's important to distinguish whether a norm has an activation condition, which describes a class of events or actions that trigger the norm's applicability. Activation conditions can be events or states of affairs.
- **Representing Obligations and Prohibitions:** Norms can generate obligations or prohibitions, depending on whether the norm designer wants agents to perform or refrain from certain actions.
- **Specific and General Deontic Relations:** Norms can generate specific deontic relations, where the debtor is known, and general deontic relations, where the debtor is not specific. General deontic relations are associated with norms that regulate actions for a set of agents, and they can result in multiple violations and fulfillment;

³In this thesis, the *norm designer* is typically an entity responsible for creating and implementing norms or rules that govern the behavior of autonomous agents within the system. The norm designer can be an individual or a group of individuals with expertise in various domains such as artificial intelligence, game theory, sociology, or computer science. Their role involves defining and encoding norms, updating or adapting norms as needed, and ensuring that the system operates effectively and ethically.

1.3. Thesis structural overview

Chapter 4 represents an extension of the methodology introduced in Chapter 3. In this paper, we introduced a rich set of norm types that could be used to study the expressive power of different formal models of norms and compare them. Moreover, we improved our previous methodology by adding two new concepts. The first one represents norms that induce the normative power among agents in the Multi-agent Systems (section 4.7.6) and the second one explains the formalization of exceptions and permissions (section 4.7.7).

Besides, our publications which focus on the main goal of my Ph.D. project, we had a chance to present an approach to formalizing the human-readable policies in the digital fashion domain into a machine-readable format. Chapter 5 discusses the increasing use of digital technologies in fashion communication and e-commerce and the associated privacy concerns. Here's a summary of the key points:

- **Digital Fashion and Privacy Concerns:** The article highlights the rapid growth of digital fashion communication and e-commerce, especially after the COVID-19 pandemic. The global fashion industry plays a significant role in the economy. However, the digital transformation of this industry raises privacy concerns for both companies and customers;
- **Customer Privacy Challenges:** One major challenge in digital fashion is customer privacy. Although companies are required to inform clients about their data policies, there is a significant gap in this area. For instance, customers are often required to accept lengthy terms and conditions before making a purchase, and a Deloitte survey in the USA found that 91% of users consent to these legal rights without reading them in this paper, we will explain how formalization of the privacy policies could improve the trust between customers and company by using an example from Zalando privacy policy notice;
- **Semantic Web Technologies and ODRL in digital fashion domain:** The article proposes using an existing policy expression language (ODRL) and Semantic Web Technologies to achieve machine-readable privacy policies.
- **Policy Monitoring and Components:** The article emphasizes the importance of policy monitoring and outlines the necessary components for a system to accomplish

automatic monitoring. It briefly references works that aim to provide solutions to this problem.

Chapter 6 concludes the thesis and proposes possible future work perspectives that could impact positively the future of the NorMAS community.

1.4 Background

1.4.1 Multi-agent Systems

Distributed artificial intelligence (DAI) has emerged as a significant subdomain within the field of Artificial Intelligence [90], with notable relevance for addressing complicated real-world challenges. Research in distributed artificial intelligence primarily includes three distinct areas [17]. These areas are *parallel AI*, *Distributed problem solving (DPS)*, and *Multi-agent Systems (MAS)*. The work that has been done in this thesis is in the field of Multi-agent Systems and more specifically Normative Multi-agent Systems which is known as NorMAS. Therefore, this section explains the key points of MAS and NorMAS.

1.4.2 Norms and Policies in Multi-agent Systems: An introduction

First of all, we would like to explain the meaning of the notion of "norm" used in this thesis. In general, a norm is defined as *"a principle of right action binding upon the members of a group and serving to guide, control, or regulate proper and acceptable behavior"*.⁴ The term "norm" has been variably utilized in diverse fields, often interchangeably with "law," "rule," "guideline," "criterion," "social expectation," and "imperative." Likewise, the definitions of normative systems vary across contexts. In the realm of Multi-Agent Systems (MAS) research, norms are put forth as formal specifications of deontic statements, serving the purpose of governing the behaviors of software agents and their interactions. To

⁴<https://www.merriam-webster.com/dictionary/norm#dictionary-entry-1>

1.4. Background

be more precise, norms find application in addressing various aspects of MAS, including coordination challenges [102], security concerns [156], modeling legal aspects in electronic institutions and electronic commerce [68], as well as depicting the structures and operations of MAS organizations [47, 83].

During the 1960s, sociologist Gibbs (1965) authored a significant paper addressing the challenges associated with defining and categorizing norms. In this work [70], Gibbs articulated that different forms of norms include the following elements:

"a collective evaluation of behavior in terms of what it ought to be; a collective expectation as to what behavior will be; and/or particular reactions to behavior, including attempts to apply sanctions or otherwise induce a particular kind of conduct."

Norms and policies play a crucial role in Multi-agent Systems. They are used to regulate the behavior and interactions of software agents within such systems. Here is an overview of their significance:

- **Norms in MAS:** In MAS, norms are explicit guidelines that establish the anticipated conduct or principles regulating interactions among agents. These norms serve the purpose of promoting efficient cooperation and coordination among agents operating within the system. Moreover, norms have a broad scope and can pertain to diverse dimensions, encompassing aspects such as coordination, security, legal considerations, and the organizational structure of the MAS;
- **Norms for Security and Coordination:** In MAS, norms play a pivotal role in tackling matters of coordination and security. These norms serve the purpose of enhancing the effective collaboration of agents while protecting the overall stability and security of the system [12];
- **Regulative and Constitutive Norms in MAS:** In MAS, regulative and constitutive norms are essential for defining and governing agent behavior. Regulative norms dictate what agents are permitted, obliged, or prohibited from doing, focusing on direct behavioral regulations like traffic rules. Constitutive norms, however, define the creation and transformation of social constructs within the system, establish-

ing the framework for agent operations by specifying the meaning and implications of actions and objects, such as defining valid transactions in a financial system. Together, these norms ensure that agents adhere to behavioral rules while understanding and navigating broader social constructs, enabling effective functioning of normative MAS [25];

- **Policies in MAS:** Policies are a broader concept that includes norms. They establish a framework of regulations, guidelines, and limitations that agents are obliged to adhere to. In contrast, norms concentrate on behavior and interactions, whereas policies extend their reach to encompass a wider spectrum of system governance, such as resource management, access control, and decision-making procedures [103].

In general, norms and policies are essential components of Multi-agent Systems. Norms define the expected behavior and govern the interactions among agents, while policies guide individual agents in making decisions and taking action. Together, they contribute to the overall functioning, coordination, and effectiveness of the Multi-agent system.

However, in this thesis, we are going to use both terms based on our use cases and consider the norms and policies of those actions that an agent should fulfill (or not violate) the deontic relations.

In addition, alongside our research field, many research groups and institutions are working on *access control policies* and, more recently, on *usage control policies*. Access control policies and usage control policies are essential mechanisms in managing and restricting access to resources within a computing environment. Access control policies focus on defining who is authorized to access specific data and under what conditions. These policies ensure data security by authenticating user credentials and allowing access based on predefined rules, such as those in Discretionary Access Control (DAC) [101], Mandatory Access Control (MAC) [154], and Role-Based Access Control (RBAC) models [Osborn2007]. They serve to manage permissions effectively, helping to mitigate the risk of unauthorized access and potential security breaches [140].

Usage control policies, on the other hand, extend the traditional access control paradigm by including ongoing controls and conditions that govern how data can be used even after

1.4. Background

access has been granted. These policies involve a combination of authorization, obligation, and condition checks that adapt over time, allowing for more granular and dynamic control over data usage. This approach is particularly important for maintaining data sovereignty and security in complex environments, as it ensures that data usage adheres to specified restrictions and compliance requirements throughout its lifecycle [141].

One of the innovative data platforms (involving over 350 companies) regarding data usage control policies is Gaia-X. We discuss Gaia-X in this section rather than in the *related works* section because Gaia-X introduces novel policy concepts that merit detailed examination here. Gaia-X is an initiative aimed at developing a federated secure data infrastructure primarily for Europe, although its principles can be applied globally. It promotes the creation of a trustworthy ecosystem where data can be shared securely while ensuring that users retain control over their data access and usage.⁵ The initiative emphasizes digital sovereignty, seeking to establish an environment where data owners have control over their information, fostering interoperability among various systems.

Gaia-X strives for innovation in the digital landscape by enabling the sharing and availability of data in a reliable and transparent manner. It aims to create an open and transparent digital ecosystem where data and services can be shared and accessed with trust and confidence. Overall, Gaia-X represents a significant European effort to enhance data sharing and collaboration while upholding values of security, sovereignty, and interoperability [151].

In Gaia-X, policies for data exchange take in various aspects aimed at defining terms and conditions for data and its exchange, reflecting a broad scope of concerns. They specifically categorized these policies into two groups *Contract Policies* and *Runtime Policies*. These policies are pivotal in establishing clear guidelines for data exchange processes and ensuring compliance with regulatory requirements. They serve as the foundation for governing data transactions between parties involved in the exchange.⁶

Contract Policies are crucial in facilitating data exchange by providing interoperable and unambiguous terms for participants to enter into agreements. These policies, which are

⁵<https://gaia-x.eu/what-is-gaia-x/>

⁶<https://docs.gaia-x.eu/technical-committee/data-exchange/22.10/dewg/#policies-for-data-exchange>

1.4. Background

both machine and human-readable, include access and usage policies and are defined using ODRL as a Policy Definition Language. *Runtime Policies*, derived from *Contract Policies*, guide the execution of contract terms within the participants' systems, ensuring seamless implementation of agreed-upon conditions. Ultimately, the negotiation and establishment of contracts through specified sequences (like IDSA⁷ or GXFS-DE⁸) lead to signed agreements, which serve as verifiable credentials describing the data asset and contractual terms.

Besides usage control policies, there exists another type of norm named as "Legal Norms". Legal norms and access control policies differ primarily in their purpose, scope, enforcement, and permanence. Legal norms are binding rules established by sovereign authorities to regulate social relations and ensure societal order, applicable to the entire society and enforced by state institutions through coercive measures, persisting even if violated.

Legal norms and access control policies differ in several key aspects. Legal norms regulate social relations broadly to ensure societal order and justice, applying to society as a whole and enforced by state authorities through legal institutions and coercive measures. They remain in effect even if violated, reflecting their binding nature. In contrast, access control policies are designed to manage and protect access to resources within specific systems or organizations. These policies apply within organizational or technological contexts and are enforced by system administrators through technological means like authentication systems and access control mechanisms. Unlike legal norms, access control policies are dynamic and can be adjusted based on organizational needs and security requirements.

Legal norms are the foundational principles and rules that govern behavior within a legal system. These norms are traditionally captured in legal texts and documents. RuleML and LegalRuleML play crucial roles in transforming these legal norms into machine-readable formats to facilitate automated processing, reasoning, and interoperability in legal and business applications.

RuleML is a standardized framework designed to represent various types of rules in a machine-readable format. It uses a wide range of rule types, such as business rules, logical

⁷https://github.com/International-Data-Spaces-Association/IDS-RAM_4_0/tree/main/documentation

⁸<https://www.gxfs.eu/specification-phase-1/>

1.4. Background

rules, and production rules. This framework establishes a foundation for rule interchange and application across different domains, thereby enabling automated reasoning and facilitating decision-making processes[114].

LegalRuleML, an extension of RuleML, is designed specifically for the legal domain. It includes formal features that address legal norms, guidelines, policies, and reasoning. LegalRuleML promotes the efficient exchange and sharing of semantic information between legal documents and software applications. This extension allows for the modeling of complex legal rules and their interrelationships, aiding in the automated processing of legal texts. Furthermore, by linking rules and provisions using Internationalized Resource Identifiers (IRI), LegalRuleML supports many-to-many relationships among legal rules and textual elements, thereby enhancing the management and reasoning of legal information [126, 16].

1.4.3 Normative Multi-agent Systems

A normative multi-agent system (NorMAS) represents a combination of models related to normative systems and Multi-agent Systems. Consequently, NorMAS is defined as a research field emerging from the intersection of normative system theory and the domain of MAS [27]. Normative systems, within the context of computer science, are known as systems in which an organization is influenced by norms, involving the inclusion of normative concepts for their description and specification [107].

Multi-agent Systems are characterized by the autonomy and self-interest of their components, each pursuing individual objectives. In particular, these systems are also recognized for their decentralized design. The significant attention garnered by Multi-agent Systems is rooted in their potential to serve as a foundational technology for complex applications that demand distributed, parallel data processing in complex and dynamic environments. Research in the field of MAS builds upon insights from distributed computing, prompting new inquiries about how agents should engage with one another to coordinate their actions and tackle complicated challenges [92].

1.4. Background

Commonly, research on MAS primarily focuses on software agents. Nonetheless, it's important to note that the agents within a Multi-agent system could just as easily be softbots or personal assistants, humans, or even teams of humans. In reality, a multi-agent system can consist of collaborative teams of both humans and software agents.

While researchers from diverse fields may have various motivations for using MAS, as highlighted in [165], the primary benefits of utilizing Multi-agent technologies include the following:

- Individual agents consider the specific context and environment of the application.
- The modeling and investigation of local interactions between agents become possible.
- Challenges related to modeling and computation are structured into sub-layers and components.

Hence, MASs offer an effective approach for managing distributed control within a computational framework. Moreover, artificial intelligence (AI) techniques can also be applied for this purpose [164].

One of the most significant advantages of using Multi-agent Systems is "*Autonomy*". Autonomy means that each individual agent independently seeks its own goals, and heterogeneity indicates that the internal states and operations of these agents may not be transparent to external parties [166]. To attain the overarching objectives of Multi-agent Systems like this, it is essential to regulate the behaviors and interactions of individual agents [43].

Two primary methods are employed to govern this regulation for individual agents. The initial approach relies on coordination artifacts, defined using fundamental coordination principles like process synchronization [132]. The second approach comes from organizational models, normative systems, and electronic institutions [143, 48]. This Ph.D. thesis falls within the domain of the second method and particularly, Normative Multi-agent Systems.

Normative multi-agent systems (NorMAS) are a specialized type of Multi-agent Systems where the behavior of individual agents is coordinated and regulated by norms. These

systems integrate concepts from both normative systems and multi-agent systems, creating a framework where agents operate based on established norms to facilitate collaboration and coordination within a group or organization. Norms in this context refer to rules or standards that dictate acceptable behavior and interactions among agents, ensuring that the collective activities align with certain objectives or social regulations [27].

The main goal of NorMAS is to enhance the efficiency and coherence of interactions among autonomous agents, particularly in environments that require complex, cooperative behavior. This is achieved by embedding normative concepts, such as obligations, permissions, and prohibitions, within the system, which agents must adhere to while pursuing their individual goals.

1.5 Semantic Web Technologies and Multi-agentsystems

In digital environments, activities are often carried out by autonomous agents who can choose to violate rules. Using Semantic Web Technologies is one of the approaches to formalize the representation of rules and monitor whether these rules are violated or not. Semantic Web technologies enhance the web to enable computers to understand, process, and utilize information available online. They aim to define and interconnect data, similar to how traditional web technologies connect web pages. Semantic technology applies formal semantics to give meaning to diverse data, making it more understandable.

The Semantic Web provides a common framework for sharing and reusing data across applications, enterprises, and communities. It attempts to bring meaning to the vast amount of information available online. Moreover, One of the goals of academic researchers is to apply their approaches at an industrial level and standard semantic web technologies could facilitate interoperability among systems, enabling seamless integration and communication between diverse applications and platforms [35].

In our method, we suggest formalizing certain aspects of norms, such as their activation conditions and the actions they regulate along with their temporal constraints, utilizing semantic web languages, specifically the W3C Web Ontology Language (OWL 2, hereafter referred to as OWL).

In this section, we describe semantic web technologies and introduce some of these technologies that have been used in our research. Moreover, we will represent the impact of the semantic web technologies on the Multi-agent system in various aspects.

1.5.1 What are Semantic Web Technologies?

The evolution of the World Wide Web has primarily revolved around serving as a platform for human-readable documents rather than data that machines can process autonomously. However, by enhancing web pages with data tailored for computer interpretation and introducing documents designed exclusively for machine understanding, we have the potential to transform the Web into what is known as the Semantic Web. The inventor of the Semantic web, Tim Berners-Lee, describes it as [24]:

“The Semantic Web is not a separate Web but an extension of the current one, in which information is given well-defined meaning, better enabling computers and people to work in cooperation.”

The core idea behind the concept of a "semantic web" is to enhance computer accessibility to the internet. Currently, the web is mainly consist of text and images, providing services primarily to human users. Computers in the existing web ecosystem have a limited role, mainly involving keyword indexing and information transfer between servers and clients. The more complex tasks, such as selection, combination, and aggregation, are left to human users. Envision a scenario where the web is optimized for machines, featuring data that is both easily readable and understandable by computers. In this thesis, we try to approach this goal and make this imagination one step closer to reality.

This enriched web would enable capabilities not feasible in the current setup. Search functionalities would evolve beyond simple keyword searches to embrace semantic elements

1.5. Semantic Web Technologies and Multi-agentsystems

like synonyms and homonyms. Moreover, it could dynamically determine the relevance of pages based on a user's current activities, eliminating the need for pre-established links applicable to all users. Information integration across various websites would be seamless, eliminating the need for users to "mentally copy and paste" data when combining information from different sources [13].

The Semantic Web, an expanded iteration of the current web, establishes a standardized framework for representing and reasoning with data. In this context, information is organized using ontologies, enhancing the capacity for drawing inferences from the stored data. The contemporary web serves three pivotal functions: searching, data combination, and data mining. The effectiveness of these tasks is intricately linked to how knowledge structures are stored and represented. Consequently, diverse schemes and languages for knowledge representation have been devised to comprehensively cover all aspects of knowledge within the domain. Key components of the Semantic Web include Ontology, RDF, OWL, and SPARQL [130].

Semantic Web technologies include a set of standards and methodologies aimed at enhancing the organization, representation, and interconnection of data on the web. Key aspects of Semantic Web technologies are:

- **RDF:** The Resource Description Framework (RDF) serves as a formal language designed for articulating structured information. Its primary objective is to facilitate the exchange of data among applications on the web, ensuring the preservation of the original meaning of the information. Unlike HTML and XML, which primarily focus on accurately displaying documents, RDF shifts its emphasis towards enabling subsequent processing and the recombination of information. This distinction positions RDF as a fundamental format for representing data in the development of the Semantic Web [79];
- **RDF Schema:** RDF Schema serves as a semantic expansion of RDF, offering tools to articulate collections of interconnected resources and the associations among them. These resources play a role in defining attributes of additional resources,

including specifications like the scope and potential values associated with properties.⁹

- **Web Ontology Language (OWL):** The Web Ontology Language (OWL) is a fundamental standard introduced by the World Wide Web Consortium (W3C) for representing knowledge on the Semantic Web. The term "ontology" finds its roots in philosophy, specifically as a sub-field dedicated to exploring the nature of existence (derived from the Greek word *Οντολογία*). Within the realm of metaphysics, this branch focuses on identifying the general categories of things that exist and developing descriptions for them. An illustration of ontological commitment is the recognition that the world comprises distinct objects, which can be categorized into abstract classes based on shared properties. In more recent times, the term "ontology" has undergone a shift due to its adoption by computer science as a clear and formal specification of a conceptualization [13]. Knowledge representation involves modeling information about real-world entities and their relationships. OWL stands out as a highly expressive, adaptable, and efficient language for knowledge representation, suitable for capturing domain-specific knowledge in areas such as healthcare, social networks, and automobiles. Derived from Description Logics (DLs), OWL ensures a well-defined meaning to the language. One notable advantage of using OWL and similar logic-based modeling languages is their support for various reasoning services. Reasoning is a method of processing background (explicit) knowledge and inferring implicit information [14].

As mentioned earlier, the main goal of the Semantic Web is to make it easier for people and machines to search, find, share, and combine information, which will accelerate the expansion of the existing Web. One thing remains consistent throughout the difficult constantly evolving nature of the Semantic Web's creation: it is gradually advancing human-software agent interactions toward intelligent content infrastructures and a more collaborative future for all.

The potential of the Semantic Web concepts for improving data generation, integration,

⁹<https://www.w3.org/TR/rdf12-schema/>

and interpretation led to their rapid adoption in the field of data and information management. In the next step, we look at the impact of these technologies on Multi-agent Systems.

1.5.2 Impact of Semantic Web Technologies on Multi-agent Systems

Semantic web technologies can play a role in supporting the representation and enforcement of norms and policies in Multi-agent Systems. Here are a few ways in which these technologies can be used for various goals:

- **Norm Representation:** Semantic web technologies, like RDF and OWL, offer a way to express rules and regulations in a format that machines can understand and organize. Norms, in this context, are outlined as ontologies, detailing classes, properties, relationships, and limitations. This structured representation enables a precise and formal definition of norms, making them evident and aiding agents in comprehending and interpreting them. This integration facilitates a more comprehensive understanding of agent organizations and their interactions within the system. [167]; This is a the responsibility of the norm designer to investigate how and where SWT can be used. For example, in [58] they used the OWL for Representation and monitoring of commitments and norms but in the T-Norm model we used the OWL for for representing the activation condition and the regulated action, and then reasoning on the state of the world.
- **Norm Interoperability:** Semantic web technologies, specifically the principles of Linked Data, allow for integrating and connecting various data sources. Likewise, in Multi-agent Systems, individual agents may own their distinct sets of norms and policies. Utilizing semantic web standards to represent norms enables agents to exchange, share, and integrate normative information. This facilitates norm interoperability, empowering agents to comprehend and adhere to norms originating from diverse sources. Moreover, Semantic Web technologies help agents engage with the

1.5. Semantic Web Technologies and Multi-agentsystems

Web of Things, promoting the creation of decentralized and open web applications within a Multi-Agent System (MAS) [42];

- **Specifying and Monitoring of Deontic Relations:** Utilizing Semantic Web languages, tools, and techniques proves beneficial in articulating and overseeing obligations within open Multi-agent Systems. This method establishes an application-independent model, increasing the normative capabilities of the system. Semantic web technologies enhance reasoning and inference capabilities, providing valuable support for enforcing norms. Agents use these capabilities to examine norm adherence, identify conflicts or inconsistencies among various norms, and draw logical conclusions using predefined normative frameworks. This process of reasoning plays a crucial role in maintaining the consistency and coherence of norms within a multi-agent system [54, 145];
- **Dynamic Norm Adaptation:** Utilizing Semantic Web languages, tools, and techniques proves beneficial for articulating and overseeing obligations within open Multi-agent Systems. This strategy establishes an application-independent model, thereby strengthening the normative capabilities of the system. In dynamic environments, Multi-agent Systems frequently require adaptations or updates to norms. Semantic web technologies offer a versatile framework for both representing and modifying norms. Agents utilize reasoning and inference capabilities to assess evolving circumstances, identify the necessity for norm adjustments, and dynamically update normative frameworks as needed.

Using semantic web technologies Systems can benefit from a standardized and structured representation of norms, norm interoperability, reasoning capabilities, and dynamic adaptation of norms. Even though using these technologies cannot solve all of our problems, it can help us to enhance the transparency and interoperability of norms and policies within the system, facilitating better coordination, cooperation, and governance among the agents.

1.6 Current Problems in Normative Systems

Today's economy is becoming more and more digital every day, with many human activities being carried out using digital devices. These digital acts generate a significant amount of digital resources, such as structured data stored in various databases, documents, photos, videos, or audio files, which users can use to perform their operations. When those digital resources are exchanged between peer agents or made accessible to data consumers, such as on the Internet, by data publishers, they may turn into digital assets.

Since the lack of a license or other agreement attached to the data is not equivalent to the ability to use it whenever someone wants to, therefore it is pointless to make digital assets available to the public without defining constraints on their usage and access. Today, Digital assets can be available under Open Data Commons licenses by adopting Creative Commons licenses (or other licenses such as Open Government License). However, it is possible that instead of using pre-defined licenses, data producers exchange digital assets with their consumers based on ad hoc agreements. This ad-hoc agreement can be a negotiation between two parties, accepting an offer, or negotiating about it.

The major problem with using this approach to constrain the access of users is that all of them are available in human-readable formats. Increasing the number of digital assets that are exchanged every day on the web, we need a formal machine-readable mechanism for the specification of licenses and agreements.

Having licenses in the human-readable format does not allow us to access some of the important features and services that can apply to digital assets. Some of these services are: (i) an advanced search of resources based on their license;(ii) the possibility to combine different resources released under different licenses by computing license compatibility or conflicts;(iii) the automatic checking of the satisfaction or violations of the normative or legal relations [81]

In order to perform the mentioned services, we need a formal syntax for representing norms in a digital format. Even though having only the syntax of a formal language is necessary, we should also represent norms using a formal semantic to reason on the constraints

1.6. Current Problems in Normative Systems

of the norms automatically.

Monitoring norms and policies in Multi-agent Systems is a challenging task that researchers and practitioners are actively working on. Here are some current problems and challenges in this area:

- **Heterogeneity of agents and dynamic environments:** Multi-agent Systems often involve agents with diverse capabilities, behaviors, and decision-making processes. They often operate in dynamic environments where agents' goals, preferences, and behavior may change over time. Monitoring norms and policies becomes complex when agents have different levels of autonomy, learning capabilities, and communication protocols. Developing monitoring mechanisms that can handle such heterogeneity is a significant challenge [153].
- **Scalability:** Multi-agent Systems can involve a large number of agents, and monitoring the behavior of each agent can be computationally intensive. As the number of agents increases, the monitoring process can become inefficient and resource-intensive. Developing scalable monitoring techniques that can handle large-scale Multi-agent Systems is a crucial problem [152].
- **Privacy of agents:** Monitoring mechanisms may require access to agents' internal state, communication, or decision-making processes, which raises concerns about privacy and security. Ensuring that monitoring respects agents' privacy while still being effective in detecting norm violations and fulfillment is a challenge. In mixed-agent groups, effective coordination among participants often requires the share of information regarding their activities, environments, and tasks. However, the decision-making process regarding what information to share and when becomes more complicated when computer agents lack access to crucial task-related information that humans possess and the privacy of this shared information should be respected carefully [65, 98, 112].

- **Dynamic Norm Adaptation:** Utilizing Semantic Web languages, tools, and techniques proves beneficial for articulating and overseeing obligations within open Multi-agent Systems. This strategy establishes an application independent model, thereby strengthening the normative capabilities of the system. In dynamic environments, Multi-agent Systems frequently require adaptations or updates to norms. The dynamic nature of open environments may necessitate the adjustment of norms. Entities acting as legislators need mechanisms and tools to identify situations where a new norm is necessary, identify redundant norms, or recognize when existing norms become invalid. Consequently, the unpredictability and dynamism inherent in open Multi-agent Systems (MAS) call for the creation of tools to identify emerging patterns of cooperation and convert them into norms [40, 112].
- **Specification of Normative Systems:** The main focus in MAS or NorMAS systems is the establishment of a normative model for modeling complex and dynamic agent societies. This model should consider various norms, supporting all types of norms, while enabling dynamic adaptation and defining mechanisms to address interrelationships among norms. Simultaneously, the development of a computational logic language is essential, requiring expressiveness for complex system definitions and tools for automated reasoning. The analysis of normative systems, including aspects like model-checking, verification, and complexity considerations, is vital for ensuring consistency and practical applicability in real-time scenarios. Additionally, the distribution of reasoning across entities is crucial for extensive applications with a multitude of simultaneous users [40].

However, the T-NORM model is not solely focused on representing digital assets. Instead, it serves as a comprehensive framework for describing and monitoring norms, such as obligations, prohibitions, and permissions, in a detailed and dynamic manner. The T-NORM model regulates the behavior of autonomous agents in an open interaction system by utilizing Semantic Web technologies such as OWL ontologies in conjunction with production rules.

1.7 Related works

In this section, we provide an overview of formal languages, frameworks, and approaches within the realms of licenses, digital rights, norms, and access control, with a specific focus on the role of Semantic Web Technologies in these domains. Even though there exists a lot of studies and researchers in these fields we decided to focus our attention on choosing the works whose goals align with the goal of this thesis.

In 2001, formal languages have been created to draft licenses and agreements, allowing for the relinquishment of certain rights related to a specific resource. Notable examples include the MPEG21-Right Expression Language¹⁰, predominantly utilized for expressing rights related to audio and video files, and ccREL [8], an RDF-based language designed for representing Creative Common licenses. MPEG21 has a major role in different standards, for example; its ability to represent contracts with ontologies which is Part ISO/IEC 21000-20 of the standard [136], to represent permissions in the IPR value chain which is Part ISO/IEC 21000-19 [134] and the newest part to do the same using blockchain technologies [100, 168].

The Open Digital Rights Language (ODRL) exhibits a broader scope compared to other languages, as it does not confine itself to a specific file type or solely focus on access control. Its resourcefulness is demonstrated by the inclusion of 126 licenses in the RD-FLicense dataset, as outlined in the ODRL 2.0 specification [135]. Originally introduced as an XML language in 2001 to express digital rights, including terms and conditions for digital content usage.

ODRL underwent significant evolution with version 2.0 in 2012, version 2.1 in 2015, and version 2.2 in 2018 [86] which helped ODRL become a more comprehensive policy language. This expansion improved the formalization of rights expressions to include the specification of privacy statements, such as duties, permissions, and prohibitions.

ODRL 2.0 and 2.1 represent a Policy Language formalized in RDF, employing an abstract model defined by an ontology. It was developed and promoted by the ODRL W3C Com-

¹⁰<https://mpeg.chiariglione.org/standards/mpeg-21/rights-expression-language>

1.7. Related works

munity Group, of which Dr. Fornara (my supervisor) is currently a chair member. In March 2016, the W3C established the "Permissions and Obligations Expression" (POE) Working Group with the objective of advancing Community Group specifications to "Recommendation" status, backed by support from the EU project Big Data Europe. While ODRL 2.1 has informal semantics outlined in English in the Core Model, there is currently no formal specification for the semantics of policies expressed in ODRL [149]. In a previous work by Kasten and Grimm in 2010, an OWL representation of ODRL 1.1 was presented, though it was limited to the representation of classes and properties without encompassing the dynamic semantics of policies, specifically how they evolve over time [97].

The ODRL Information Model 2.2 was released on February 15, 2018. This release aimed to provide a more flexible and interoperable framework for expressing digital policies. The update included enhancements to the underlying semantic model, vocabulary, and formal semantics, allowing for clearer definitions and better implementation practices of digital rights management and policy expression.

The primary reason for releasing ODRL 2.2 was to address limitations and incorporate feedback from the previous version, ODRL 2.1. The update aimed to improve the expressiveness and usability of the language, ensuring that it could effectively handle a wider range of digital content usage scenarios. By refining the information model and expanding the vocabulary, ODRL 2.2 helps organizations manage digital rights more efficiently and accurately.¹¹

A great limitation of ODRL is not having an operational semantics that allows it to compute the fulfillment or the violation of policies. Compared to the T-Norm Model, ODRL does not allow for the specification of generic policies that apply to different agents each time. In ODRL, the target of a policy must always be a specific agent. However, in our abstract model for policy specification, it is possible to apply one policy to all the agents who will perform a certain action or who plays a certain role.

In addition, we treat exceptions (especially permissions) differently from obligations and

¹¹<https://w3c.github.io/odrl/bp/>

prohibitions. In our view, exceptions are derived from basic constructs that define obligations and prohibitions.

Finally, while ODRL uses a fixed set of deontic types, our model focuses on detailing the sequence of events that bring to the violation or fulfillment of a policy. A key similarity between the T-NORM and ODRL models is their use of semantic web technologies. However, they use them differently, ODRL use the OWL language to define the policy meta-model, while the T-NORM (and also OWL-POLAR models) OWL to model actions by agents, as well as to express activation conditions and actions regulated by norms.

The L4LOD (Licenses for Linked Open Data) is a concise vocabulary designed to express licenses for Linked Open Data (LOD), as introduced in [75]. The semantic framework of this language can effectively express the deontic aspects of licenses by extending Defeasible Logic [115]. Defeasible Logic, a non-monotonic logic, helps the representation of obligations as defeasible rules, constraining actions as obligations to refrain from certain activities, and a basic concept of permission defined as defeaters of obligations (i.e., defeaters of prohibitions) [119].

The primary objective of this research is to automate the determination of the deontic elements within a composite license (denoted as lc), derived from a set of individual licenses $l_1, \dots, l_n$. This process involves the application of deontic rules, categorized as strict rules, defeasible rules, and defeater rules. Additionally, two heuristics for OR-composition and AND-composition are employed to combine the deontic effects of licenses. The formalized content of these licenses is expressed through propositions such as ShareAlike, Attribution, or Commercial. This approach, grounded in non-monotonic logic, is particularly noteworthy in cases where the regulated actions lack specific attributes such as actors, contexts, or relevant parameters (like an amount of money for paying actions).

In the field of studies on Access Control to data, an important XML standard language is the eXtensible Access Control Markup Language XACML [113] [133] which focuses on the specification of access control policies. Recognized as an OASIS industry standard, XACML serves as a crucial application-independent policy language. Primarily operating as an Attribute-Based Access Control system, it uses an XML-based language to articulate

1.7. Related works

and exchange access control policies across the Web. XACML describes both a policy language and an access control decision request/response language.

Policies within XACML are characterized by subjects, resources, environments, and actions. The subject element represents the entity seeking access, possessing one or more attributes. The resource element can be data, services, or system components, also featuring one or more attributes. The action element expresses the type of access requested on the resource through one or more attributes. Additionally, an environment element can optionally provide supplementary information.

While the formalization of obligations is a significant objective in this thesis, the XACML standard has limited support in this regard. Specifically, XACML only provides a general syntax for obligation specification, with its primary focus on permissions and prohibitions. The language's core functionalities deal with access control, with provisions for standard extension points to define new functions, attributes, policy combination logic, and more. In addition to the language, XACML outlines an architecture for policy evaluation and a communication protocol for message interchange. It also supports role specification, multi-valued attributes, policy indexing, and the specification of sticky policies.

Among all those rights, policies, and access control languages, ODRL 2.2 is the more general which allows the specification of licenses but it also has more customizable offers of digital assets by data providers, requests of digital data by data consumers, or agreements that binds two parties on a set of permissions and a prohibition on a digital asset. Moreover, it has another advantage, given that it is the starting point of a W3C working group, it will likely become an important and well-known standard among Web developers in the next years.

Solid is a framework for decentralized personal data storage, called Pods, designed to enhance user control and interoperability using linked data and web standards.¹² The data within these Pods is managed through the Web Access Control (WAC) specification, which uses Internationalized Resource Identifiers (IRIs) to represent resources and agents. Authorization statements for accessing data are stored in an Access Control List (ACL) for

¹²<https://solidproject.org/>

1.7. Related works

each resource or inherited from its parent resource. WAC currently supports four operations: read, write, append, and control, with access permissions specified for individual agents or trusted applications. By default, all operations are prohibited unless explicitly allowed by an ACL authorization, putting users in charge of who can access their data.

Given the individual control Solid provides over personal data, it is crucial to consider the implications of data protection laws like the European General Data Protection Regulation (GDPR). GDPR mandates that data controllers disclose specific information, including identity, purpose, data categories, legal bases, and recipients, whether data is collected directly or indirectly. This transparency helps individuals make informed decisions about their personal data. Typically, this information is conveyed through notices, consent dialogues, and privacy policies [50].

In [49], represents how Solid can achieve this transparency through conventional methods such as website notices, with authorizations stored in the ACL. However, to fully empower users, Solid Pods should also contain this information. This allows users to introspect on their data usage, store consents for accountability, set data-sharing preferences, and manage consent withdrawal. Understanding legal requirements, user information needs, and desired control forms is essential for achieving this. The authors address these goals by exploring how Solid's ACL can be extended to include and enforce user data-sharing preferences using Open Digital Rights Language (ODRL) policies and the Data Privacy Vocabulary (DPV) [128] for legal compliance. The proposed mechanism aims to automate consent requests and manage permissions effectively while discussing practical challenges and solutions.

The studies on Normative Multi-agent Systems (NorMAS), concern mainly the formalization of norms used for expressing obligations, permissions, and prohibitions, and the definition and the realization of fundamental functionalities for norms definition, monitoring, enforcement, and for norms adoption and reasoning.

Currently, within the Multi-agent community, there are ongoing efforts that use Semantic Web Technologies for the formalization and administration of norms or policies. Notably, the works of Fornara's group explore the utilization of OWL 2, SWRL rules, and OWL-

1.7. Related works

API to specify commitments and obligations of agents [58, 54]. These papers introduce an OWL ontology representing obligations as a class of potential actions to be completed within specified deadlines. Monitoring the fulfillment or violation of these obligations, based on agents' actions, requires a dedicated framework managing time elapse and conducting closed-world reasoning on specific classes.

Unfortunately, the scalability of this approach poses limitations, hindering its practical use in real applications. A similar ontological formalization of obligations has been extended for application in a broader OWL 2 model of artificial institutions, instantiated at runtime through the dynamic creation of interaction spaces [64]. In [116] and [117], the OWL 2 formalization of obligations is applied to regulate activities of social scientists working with data collected from the Web, ensuring compliance with legal and ethical standards. Lastly, the OWL 2 formalization of obligations is adapted for specifying and managing obligations of data consumers and providers, regulating data access formalized in OWL 2 [61, 104].

Another interesting approach employing Semantic Web Technologies for formalizing and managing norms is the OWL-POLAR framework for semantic policy representation and reasoning [145]. This framework explores the use of OWL ontologies to represent the interaction state among agents and utilizes SPARQL queries for reasoning on policy activation, predicting potential conflicts among policies, and resolving conflicts.

A notable distinction from the methods proposed by Fornara's group is that in the OWL-POLAR model, where the activation conditions and policy content (prohibitions, permissions, or obligations) are not solely expressed using the OWL language. Instead, they are represented using conjunctive semantic formulas—combinations of atomic assertions using concepts and relations defined in an OWL ontology. Consequently, the OWL-POLAR framework requires translating activation conditions and policy content into SPARQL queries for evaluation on the OWL ontology representing the state of the world. In contrast, Fornara's group expresses activation conditions and obligation content using OWL classes, allowing direct reasoning through querying the OWL ontology for individuals belonging to the defined activation condition and content classes. Notably, OWL-POLAR

lacks a treatment of time. One noteworthy contribution of the OWL-POLAR framework is its examination of conflicts among norms, proposing conflict resolution. This framework manages norms whose content is represented as action instances using concepts and properties from the state ontology. The reasoning mechanisms presented are decidable, though the use of OWL-DL may render them intractable in the worst case. Reducing expressiveness to DL-Lite ensures decidability.

Another noteworthy proposal using Semantic Web technologies for policy specification and management is the KAoS policy management framework [157, 33]. This framework comprises three layers: (i) the human interface layer, where policies expressing authorizations and obligations are articulated in constrained English sentences; (ii) the policy management layer, utilized to encode policy-related information in OWL; and (iii) the policy monitoring and enforcing layer, employed to compile OWL policies into an efficient format for monitoring and enforcement. Additionally, another approach partially utilizing Semantic Web technologies is the Rei policy language [94], which models deontic concepts such as rights, prohibitions, obligations, and dispensations. Rei is implemented in the logic language Prolog and incorporates some ontologies enabling the policy engine to interpret a subset of RDFS policies.

An interesting approach for representing the Legal Norms using OWL ontology is represented in [71]. This paper presents a formal model of legal norms using the Web Ontology Language (OWL) to facilitate semiautomatic drafting, semantic retrieval, and browsing of legislation. Unlike traditional models based on formal logic, rules, or ontologies that do not support legislative drafting and retrieval, this model precisely defines legal norms and their relationships. It integrates these norms with the XML model of legislation according to the CEN MetaLex specification, addressing issues such as legal rule fragmentation that hinder effective usage of legislation. The model's effectiveness is validated through application to existing legislation and a prototype for semantic browsing, aiming to improve the quality and usability of legal drafting and retrieval processes.

It is evident that despite extensive exploration of norms and policies in multi-agent research, there is considerable room for improvement in existing models and languages for

1.7. Related works

norm and policy formalization. This improvement involves evaluating their expressivity in real-world use cases, enhancing the performance and scalability of the framework needed for their management (creation, enforcement, and monitoring), and ensuring the practical usability of these approaches for norm designers and industry.

In the realm of Access Control research, notable works present models for specifying access control policies and their enforcement mechanisms when a user submits a data request, particularly focusing on the specification of policies and protected data using Semantic Web Technologies. In [52] a Role Based Access Control model, called ROWL-BAC, is formalized in OWL using two different possible approaches for expressing permissions and prohibitions in terms of the roles played by the user who wants to perform a certain action. In particular, the differences between formalizing roles as classes and roles as values are discussed in great detail.

In [139], a framework is presented for managing permissions in the Web of Data, including read, write, and attribute-based permissions. They propose the Privacy Preference Ontology (PPO) for Linked Data, which aims to control access to RDF statements, named-graphs, and resources. The PPO enables users to define access test queries using SPARQL ASK queries, specifying criteria that users must meet to be granted access. These queries assess the FOAF profile of the requester to determine if they have access to other users' FOAF profiles. This framework operates on a private-by-default policy, meaning resources without specified privacy preferences are inaccessible.

In [39] a context-based authorization mechanism for RDF graph is presented. Since they aim at mobile devices, their model is grounded on two ontologies, one for access control policies and another for the mobile context. Similarly to [139], they propose to use SPARQL 1.1 ASK queries on the context of the user for expressing access conditions.

In [129], a content-based access control policy is defined by using a list of critical tags or keywords that social network users specify in order to express which information they want to protect from certain roles. This may be classified as a semantic approach because the list of keywords provided by the user of the social network is enriched using Linked Data to look for related concepts.

1.7. Related works

In [34], they explain how the rapid digitalization of the world has led to significant economic and social impacts, further underscored by the COVID-19 pandemic, which highlighted the importance of digital tools in managing privacy, surveillance, transparency, and censorship. This context raises critical questions about how nations handle extensive digitalization and control the technologies and applications used within their borders. Europe has taken a proactive approach to this challenge with the launch of GAIA-X, an initiative stemming from the German Federal Government, aimed at creating a next-generation data infrastructure that ensures digital sovereignty, fostering innovation while adhering to European regulatory standards. The GAIA-X initiative is designed to establish a trusted ecosystem where data and services can be securely exchanged and shared.

As mentioned before, GAIA-X aims to differentiate itself from existing platforms like GAFAM and BATX by focusing on data sovereignty and compliance with European regulations. The initiative leverages the International Data Spaces Association (IDSA) architecture, which was established in 2019 to facilitate secure data exchange within business ecosystems. The IDSA framework ensures that data owners retain full control over their data, defining usage constraints to maintain security and trust. By incorporating these principles, GAIA-X provides a federated and interoperable infrastructure that supports the sovereign exchange of data. This approach is intended to create a more resilient digital environment in Europe, reducing dependency on non-European technologies and enabling a diverse ecosystem of actors to innovate and collaborate securely.

Recently, researchers introduced various approaches using Large Language Models. In [89], they discuss how Large Language Models (LLMs), such as GPT-4, can help automate the process of turning legislative text into structured formats. This step is crucial for developing AI-based legal systems, which assist people in understanding how laws apply to them by providing clear information. Traditionally, encoding laws in these structured formats is a slow process, creating challenges in building these systems efficiently. The study uses LLMs to generate legal pathways based on the JusticeBot system. The results are encouraging, showing that 60% of the pathways created by LLMs are as good as, or better than, those made by humans. This suggests that LLMs could significantly speed up the creation of transparent and easily understandable legal systems. However, LLMs

1.7. Related works

are not without challenges. Issues such as bias, misinformation, and the difficulty of ensuring fairness in decision-making remain key concerns when applying LLMs to norm governance.

2

A Framework for Automatic Monitoring of Norms that Regulate Time Constrained Actions

2.1	Abstract	38
2.2	Introduction	38
2.3	Design Goals	41
2.4	The T-Norm model of norms	42
2.4.1	Expressive power of the model	47
2.4.2	The model of Exceptions	50
2.5	Operational Semantics of the model	53
2.6	Architecture of the framework and its implementation	56
2.7	Related Work	59

Published Manuscript: first co-authored with Dr. Nicoletta Fornara and Prof. Marco Colombetti: Fornara, N., Roshankish, S., Colombetti, M. (2022). A Framework for Automatic Monitoring of Norms that Regulate Time Constrained Actions. In: Theodorou, A., Nieves, J.C., De Vos, M. (eds) Coordination, Organizations, Institutions, Norms, and Ethics for Governance of Multi-Agent Systems XIV. COINE 2021. Lecture Notes in Computer Science, vol 13239. Springer, Cham. <https://doi.org/10.1007/978-3-031-16617-42>

2.1 Abstract

This paper addresses the problem of proposing a model of norms and a framework for automatically computing their violation or fulfillment. The proposed model can be used to express abstract norms able to regulate classes of actions that should or should not be performed in a temporal interval. We show how the model can be used to formalize obligations and prohibitions and for inhibiting them by introducing permissions and exemptions. The basic building blocks for norm specification consist of rules with suitably nested components. The activation condition and the regulated actions together with their time constraints are specified using the W3C Web Ontology Language (OWL 2). Thanks to this choice, it is possible to use OWL reasoning for computing the effects that the logical implication between actions has on the fulfillment or violation of the norms. The operational semantics of the model is specified by providing an unambiguous procedure for translating every norm and every exception into production rules.

2.2 Introduction

In this paper, we present the T-Norm model (where T stands for Temporal), a model for the formalization of *relational norms* that regulate classes of actions that agents perform in the society and that put them in relation to other agents, like for example paying, lending, entering a limited traffic area, and so on. Our proposal is strictly related to the specification of the operational semantics of such a model to make it possible to provide monitoring and simulation services on norms specifications. Specifically, the proposed model can be used to automatically compute the fulfillment or violation of active obligations and prohibitions formalized to regulate a set of actions that should or should not be performed in a *temporal interval*. The fact that the actions regulated by the norms are time constrained is an important distinguishing feature of the proposed model. We would like to stress that the type of norms that can be represented with the proposed model are assumed to be *static*, in the sense that they do not change dynamically over time. What is represented in the model is that a norm may be activated over time and subsequently its activation

2.2. Introduction

may generate fulfilments and/or violations. Another important aspect of the model is that once a set of obligations and prohibitions are formalized they may be further refined by the definition of permissions and exemptions.

In NorMAS literature [159] it is possible to find numerous formal models for the specification of norms, contracts, commitments and policies. Many of them can be used for regulating the performance of actions or for requiring the maintenance of certain conditions, but very often those actions and conditions may only be expressed using propositional formulae. This choice makes it difficult to express the relation between the regulated actions (or conditions) and time. This is an important limit in the expressiveness of those models, because there are numerous real examples of norms and policies whose relation with time intervals is important for their semantics; for example in e-commerce, deadlines (before which a payment must be done) are fundamental for computing the fulfilment or violation of contracts.

When temporal aspects are important for the specification of norms and for reasoning on the evolution of their normative state, one may decide to use temporal logics (e.g. Linear Temporal Logic LTL) to express and reason about time-related constraints. Unfortunately, this solution has important limitations when it is necessary to use automatic reasoning to compute the time evolution of the normative state, as discussed in [127].

In our approach we propose to formalize some components of the norms, i.e. their activation condition and the regulated actions together with their time constraints using semantic web languages, specifically the W3C Web Ontology Language (OWL 2, henceforward simply OWL)¹. This is for two reasons. First, it should be easier for those who want to formalize norms with our model to use a standard language that is fairly well known and taught in computer science graduate courses. Second, this language has a formal semantics on which automatic reasoning can be performed. Moreover, OWL is more expressive than propositional formulae², which are used in many other norm models. The idea of formalizing policies using semantic web languages is spreading also thanks to the success of

¹<https://www.w3.org/TR/owl2-overview/>

²Description Logics (DLs), which are a family of class (concept) based knowledge representation formalisms, are more expressive than propositional logic, and they are the basis for ontology languages such as OWL [82].

2.2. Introduction

the ODRL (Open Digital Rights Language) policy expression language³, which has been a W3C Recommendation since February 2018.

Our idea is to propose a model of norms that norm designers can use for formalizing the intuitive meaning of having an obligation or a prohibition. That is, when something happens and certain conditions hold, an agent is obligated or prohibited to do something in a given interval of time. What is innovative with respect to other approaches is that instead of explicitly labelling a norm as an obligation or a prohibition, we let the norm designer explicitly express what sequence of events will bring to a violation or a fulfilment; this way, there is practically no limit to the types of normative relations that can be expressed. To do so the norm designer needs to be able to describe triggering events or actions and their consequences. The resulting model will let the norm designer to specify norms as *rules* nested into each other.

The main contributions of this paper are: (i) the definition of a model of norms that can be used to specify numerous types of *deontic relations*, i.e. different types of obligations and prohibitions, and their exceptions; (ii) the definition of its operational semantics (by combining OWL reasoning and forward chaining) that can be used to automatically monitor or simulate the fulfilment or violation of a set of norms; (iii) the proposal of a set of different types of concrete norms, that can be used to evaluate the expressive power of a norms model.

This paper is organized as follows: in Section 2.3 the main goals that guided the design of the norms model are presented. In Section 2.4 the T-Norm model is introduced and in Section 2.5 its operational semantics is provided. In Section 2.6 the architecture of the framework for computing the fulfilment or violation of norms and its implementation are presented. Finally, in Section 2.7 the proposed model is compared with other existing approaches.

³<https://www.w3.org/TR/odrl-model/>

2.3 Design Goals

In this section we list the main goals that guided us in the design of the proposed model and in the definition of its operational semantics.

Our first goal is to propose a model of norms able to regulate *classes of actions*; for example, we want to be able to formalize a norm that regulates all the accesses to a restricted traffic zone of a metropolis and not only the access of a specific agent. This objective is not achieved by those models, like ODRL and all its extensions or profiles [60, 44] where the policies designer has to specify the specific debtor of every policy instance.

Our second goal is to define a model of norms able to regulate classes of actions whose performance is *temporally constrained*. For instance, the following norm regulates all the access to an area and the subsequent action of paying is temporally constrained by a deadline, which in turn depends on the access instant of time: “*when an agent enters in the limited-traffic area of Milan, between 7 a.m. and 7 p.m., they have to pay 6 euros within 24 hours*”. The first and the second goal will bring us to define a model for expressing norms that may be applied multiple times to different agents and may be activated by numerous events happening in different instants of time.

Starting from the experience that we gained by using the model of obligations, prohibitions and permissions presented in our previous paper [60], we have developed a third goal for our model. The goal is to propose a model made of basic constructs that can be combined by a norm designer to express different types of deontic relations without the need to introduce a pre-defined list of deontic types, like obligation, prohibition, and permission. This has the advantage that whenever a new kind of norms is required, like for example the notion of exemption or the notion of right, there is no need to introduce a new type into the model with its operational semantics. With the model proposed in this paper, it is possible to use few basic constructs and combine them in different ways to express the obligation to perform an action before a given deadline or the prohibition to perform an action within an interval of time. Our idea is to allow norm designers to explicitly state what behaviour will bring to a violation and what behaviour will bring to a fulfilment, regardless of whether they are formalizing an obligation or a prohibition. Moreover, in

this new model, permissions are not treated any more as first-class objects, but they are formalized as exceptions to prohibitions, while exemptions are formalized as exceptions to obligations.

Our fourth goal is to provide an operational semantics of our model of norms that will make it possible to *monitor* or *simulate* the evolution of their state. Our goal is mainly to be able to automatically compute if a policy is *active* (or in-force) and then if it becomes *fulfilled* or *violated* on the basis of the events and actions performed by agents. Monitoring is crucial from the point of view of policy's *debtor* for checking if their behaviour is compliant and it is relevant for policy's *creditors* to react to violations. Simulation may be used for evaluating in advance the effects of the performance of certain actions. Another useful service that can be provided on a set of policies is checking their consistency, checking for example if a given action is contemporarily obligatory and prohibited. This can be done at design time by using model-checking techniques, but it is not among the objectives of our model. However, by proposing a model that allows us to track how the state of the norms evolves over time, it will be possible to detect inconsistencies that occur at a precise instant of time during the execution of their monitoring process.

2.4 The T-Norm model of norms

The idea that has guided us in the definition of the T-Norm model is to give norm designers a tool to describe what sequence of events or actions would bring an agent to the violation or fulfilment of a norm. This approach has the advantage of providing norm designers with a model that in principle can be used to define any type of deontic relationship, like obligations, prohibitions, permissions, exemptions, rights and so on. This is a crucial difference with respect to the models having a pre-defined set of deontic types, like it is the case for ODRL, OWL-POLAR [145], and also our previous proposal of a model for monitoring obligations, prohibitions, and permissions [60].

The intuitive meaning of having an obligation (resp. a prohibition) that we want to capture is the following one: when an activation event happens and some contextual conditions

2.4. The T-Norm model of norms

are satisfied, it is necessary to compute some parameters (for example the deadline) and to start to monitor the performance of a specific regulated action or class of actions. In turn, if an action, which matches the description of the regulated one, is performed before another event (for example a time event that represents a deadline), then the obligation is fulfilled (resp. the prohibition is violated); otherwise, if the regulated action cannot be performed anymore (for example because the deadline has elapsed) the obligation is violated (resp. the prohibition is fulfilled).

To capture this intuitive meaning we decided to represent norms as rules that determine the conditions under which a fulfilment or violation is generated. Below, we discuss how different types of norms can be represented in this way. Then in the next section we describe how T-Norm norms can be translated into production rules, thus assigning an unambiguous operational semantics to our norm formalism.

The idea of representing norms in the form of rules is not new in NorMAS literature [68]. However, in order to explicitly specify the sequence of events that bring to a violation or a fulfilment, we propose a model of norms where the basic building blocks for norm specification consists of rules with properly nested components. Thanks to this choice, as we will discuss in Section 2.5, the operational semantics of our model of norms can be easily expressed using productions. Our idea is to express the meaning of having an obligation or a prohibition by nested rules of the following form:

```
NORM Norm_n
[ON ?event1 WHERE conditions on ?event1
THEN
  COMPUTE]
  CREATE DeonticRelation(?dr);
  ASSERT isGenerated(?dr,Norm_n); [activated(?dr,?event1);]
  ON ?event2 [BEFORE ?event3 WHERE conditions on ?event3]
    WHERE actor(?event2,?agent) AND conditions on ?event2
  THEN ASSERT fulfills(?agent,?dr); fullfilled(?dr,?event2)|
    violates(?agent,?dr); violated(?dr,?event2)
```

2.4. The T-Norm model of norms

```
[ELSE ASSERT violates(?agent,?dr); violated(?dr,?event3) |  
      fulfills(?agent,?dr); fulfilled(?dr,?event3]
```

In the proposed model the first (optional) ON...THEN component is used for expressing *conditional norms*, i.e. norms that start to be in force when a certain event happens and where the *temporal relation* between the activating event and the regulated action is crucial in the semantics of the norm. For example in Norm01 (“*when an agent enters in the limited-traffic area of Milan between 7 a.m. and 7 p.m., they have to pay 6 euros within 24 hours*”) the event of entering in the limited-traffic area must occur for the obligation to pay to activate; moreover the entering instant is fundamental for computing the deadline of the payment. The second ON...THEN component is used for expressing the actions regulated by the norm and the consequences of their performance or non-performance.

In the T-Norm model, a norm activation can generate many different *deontic relations*. In other approaches, like for example in [11], a norm generates norm instances. We prefer to use the term deontic relation because it can also be used to denote obligations and prohibitions that are not created by activating a norm, but for example by making a promise or accepting an agreement.

In the ON ?event WHERE and in the BEFORE ?event WHERE components, the norm designer has to describe the *conditions* that a real event has to satisfy to match a norm. In our model all the relevant events and actions are represented in the *State Knowledge Base*. The data for managing the evolution of the state of norms, for example, the deontic relation objects, are stored in the *Deontic Knowledge Base*. Obviously, the formalism chosen for representing the data in the *State KB* and the *Deontic KB* determines the syntax for expressing: the *conditions*, which are evaluated on the *State KB*, and the *actions* (after THEN), which are performed on the *Deontic KB*. Differently from other approaches, where the context or state of the interaction is represented by using propositional formulae [11, 75], we decided to formalize the *State KB* and the *Deontic KB* by using semantic web technologies, in particular the W3C Web Ontology Language (OWL). This choice has the following advantages:

2.4. The T-Norm model of norms

- events and actions are represented with a more expressive language, indeed OWL is a practical realization of a Description Logic known as *SROIQ(D)*, which is more expressive than propositional logic;
- in the definition of the conceptual model of the State KB it is possible to reuse existing OWL ontologies making the various systems involved in norm monitoring interoperable;
- it is possible to perform automatic reasoning (OWL can be regarded as a decidable fragment of First-Order Logic) on the *State KB* and deducing knowledge from the asserted one. In particular, this is important when the execution of an action logically implies another one. For example, the reproduction of an audio file implies its use, therefore if the use is forbidden so is its reproduction. This is a crucial advantage because instead of creating special properties that allow you to express which actions imply other ones (like for example it has been done in ODRL with the *implies* property⁴) it is sufficient to reason on the actions performed by using OWL reasoners.

In the specification of *conditions*, OWL classes are represented using unary predicates starting with a capital letter and OWL properties are represented using binary predicates starting with a lowercase letter. If an event is described with more conditions, they are evaluated conjunctively, variables (starting with ?) are bound to a value, and a negated condition is satisfied if there is no element in the KB that matches it. In the example reported in this paper, the conceptual model of the events represented in the *State KB* is formalized with the *Event Ontology* in OWL [54, 60], which imports the *Time Ontology* in OWL⁵ used for connecting events to instants or intervals of time, and the *Action Ontology* for representing domain-specific actions, like the *PayAction* class. The conceptual model of the *Deontic KB* is formalized with the *T-Norm Ontology* in OWL⁶.

In the second component of norms, the BEFORE condition and the ELSE branch are optional. The BEFORE part is mainly used for expressing deadlines for obligations. Although

⁴<https://www.w3.org/TR/odrl-model/#action>

⁵<https://www.w3.org/TR/owl-time/>

⁶<https://raw.githubusercontent.com/fornaran/T-Norm-Model/main/tnorm.owl>

2.4. The T-Norm model of norms

an obligation without a deadline cannot be violated and therefore it is not an incentive to perform the obligatory action, the BEFORE part is not compulsory. The ELSE branch is followed when the regulated action cannot happen anymore in the future, for example if it has to happen before a given deadline (in this case event3 is a time event) and the deadline expires without event2 being performed. In principle other conditions, beside BEFORE, can be used to express other temporal operators but they are not introduced in this version of our model. In the consequent (THEN) parts of a norm, it is possible to specify that:

- (COMPUTE) the value of some variables (for example the deadline that depends on the activation time) are computed using arithmetic operations and the value of variables obtained when matching the antecedent;
- (CREATE) new individuals belonging to a certain class and having certain properties have to be created in the *Deontic KB* for making the monitoring of norms feasible. Each conditional norm, when activated, can generate several deontic relations;
- (ASSERT) the value of certain properties of existing individuals created by the norm may be set.

The *debtor* of a deontic relation is the agent that is responsible for its violation or fulfilment; usually it is the actor of the regulated action. In legal systems there are exceptions to this general rule (for example for actions performed by minors or people with mental impairment or in cases of strict liability), but we leave this aspect for future works. Specifying the debtor is important because it is the agent to whom sanctions will apply (this aspect is not addressed in the current paper).

The *creditor* of a deontic relation is the agent to whom the debtor owns the performance or non-performance of the regulated action. In certain cases it may be difficult to establish the creditor of a deontic relation (for example who is the creditor of the prohibition of running a red light?); we leave for future works the analysis of this aspect.

2.4.1 Expressive power of the model

By using the T-Norm model we are able to express different types of norms. First of all it is possible to formalize conditional and direct (or un-conditional) obligations and conditional and direct prohibitions. Moreover, every conditional norm (whether it is an obligation or a prohibition) when activated will bring to the creation of *specific deontic relations* or to the creation of *general deontic relations*. Every specific deontic relation regulates the performance of an action by a specific agent, differently every general deontic relation regulates the performance of a class of actions that can be concretely realized by different agents, and therefore can generate many violations or fulfilments.

There exist models, which are not focused on regulating time-constrained actions, where (coherently with deontic logics) prohibitions are merely formalized as obligations to not perform the regulated action. However, when the regulated actions are time constrained it is crucial to react to their performance but also to their non-performance in due time. Think for example to the prohibition expressed by Norm02: “*Italian libraries cannot lend DVDs until 2 years are passed from the distribution of the DVD*”. This prohibition cannot be expressed as an obligation to not lend certain DVDs in a specific time interval, because while an obligation is fulfilled by the performance of a single instance of an action, a prohibition is not fulfilled by a single instance of refraining from the performance of an action. In the T-Norm model the main difference between obligations and prohibitions is that the performance of the regulated action brings about a fulfilment in the case of obligations and a violation in the case of prohibitions.

To illustrate the flexibility of our model, we shall now present examples of different types of norms. Due to space limitation we will not formalize them all. Norm01 is an example of *conditional obligation* and each one of its activations creates one *specific* deontic relation to pay 6 euros for the owner of every vehicle that entered in the limited-traffic area. Norm01 may be formalized with the T-Norm model in following way:

NORM Norm01

ON ?e1

WHERE RestrictedTrafficAreaAccess(?e1) AND vehicle(?e1,?v) AND

2.4. The T-Norm model of norms

```
owner(?v,?agent) AND atTime(?e1,?inst1) AND
inXSDDateTimeStamp(?inst1,?t1) AND ?t1.hour>7 a.m. AND ?t1.hour<7p.m
THEN
COMPUTE ?t_end.hour=?t1.hour+24
CREATE DeonticRelation(?dr01_n);TimeEvent(?tev_end_n);
Instant(?inst_end_n);
ASSERT isGenerated(?dr01_n,Norm01);activated(?dr01_n,?e1);
debtor(?dr01_n,?agent);end(?dr01_n,?tev_end_n);
atTime(?tev_end_n,?inst_end_n);
inXSDDateTimeStamp(?inst_end_n,?t_end);
ON ?e2 BEFORE ?tev_end_n
WHERE PayAction(?e2) AND reason(?e2,?e1) AND recipient(?e2,Milan)
AND price(?e2,6) AND priceCurrency(?e2,euro) AND actor(?e2,?agent).
THEN ASSERT fulfills(?agent,?dr01_n); fulfilled(?dr01_n,?e2)
ELSE ASSERT violates(?agent,?dr01_n); violated(?dr01_n,?tev_end_n)
```

where a counter *n* is incremented each time the norm is activated, so that each activation creates a different deontic relation.

Norm02 (“*Italian libraries cannot lend DVDs until 2 years are passed from the distribution of the DVD*”) is an example of *conditional prohibition*, its activation creates a *general* deontic relation every time a new DVD is distributed. The general deontic relation created for each specific DVD regulates the actions of all the agents registered in Italian libraries.

```
NORM Norm02
ON ?e1
WHERE isReleased(?e1) AND object(?e1,?dvd) AND VideoObject(?dvd)
AND place(?e1;Italy) AND atTime(?e1,?inst1) AND
inXSDDateTimeStamp(?inst1,?t1)
THEN
COMPUTE ?tend_n=?t1.year+2
CREATE DeonticRelation(?dr);TimeEvent(?tevend_n);Instant(?instend_n);
```

2.4. The T-Norm model of norms

```
ASSERT  isGenerated(?dr, Norm02); activated(?dr, ?e1);
        end(?dr, ?tevend_n); atTime(?tevend_n, ?instend_n);
        inXSDDateTimeStamp(?instend_n, ?tend_n);
ON      ?e2 BEFORE ?tevend_n
        WHERE LendAction(?e2) AND object(?e2, ?dvd) AND actor(?e2, ?agent)
THEN ASSERT violates(?agent, ?dr); violated(?dr, ?e2)
```

In Norm02 we can specify the ELSE actions for computing the fulfilment of the prohibition only when it is useful for management needs and by taking into account that there will be many more fulfilments than violations. This is because, given that Norm02 generates general deontic relations, each deontic relation will be fulfilled by all the agents who did not violate it.

The third type of norms is a *conditional prohibition* that generates *specific* deontic relations, an example of this type of norm is Norm03: “*a person who has a positive swab to Covid-19 cannot leave the house for the next 15 days*”. The fourth type of norm is a *conditional obligation* that generates *general* deontic relations, like for example in Norm04: “*when the school bell rings, students have 5 minutes to enter their classroom*”.

An example of *unconditional prohibition* is given by Norm05: “*when the red light is on it is prohibited to pass the traffic light*”. This prohibition is unconditional because there is no need to react to its activation by performing specific actions. For enforcing this prohibition it is enough to check the state of the red light every time an agent pass the traffic light and if the red light is on there is directly a violation. Finally, the following Norm06: “*the lecturer of a course has to organize 2 exams per year*” is an example of *unconditional obligation*.

Finally, we present an example of conditional obligation where the obliged action should be performed before another event that is not a time event (there is not a deadline). Norm07 is: “*When an agent enters into a supermarket parking between 7 a.m. and 7 p.m., they have to pay 2 euros for every hour of the parking unless they did some shopping at the supermarket*”.

It is important to mention an important constraint for the use of the model: the regulated action must have an actor, this actor is the debtor of the deontic relation and is the agent who will fulfill or violate the deontic relation.

2.4.2 The model of Exceptions

The meaning of having the *permission* to perform an action has been widely studied in the literature and different types of permission have been analyzed. In [73] the important distinction between *strong* and *weak* permission has been discussed. Having the weak permission to do an action is equivalent to the absence of the prohibition to do such an action. Differently, we have the strong permission to do an action when there is the explicit permission to do such an action; usually strong permissions are used to explicitly derogate to existing prohibitions. A similar notion is that of *exemption*, which is used to derogate obligations. In the T-Norm model we introduce one construct, the *exception*, that can be used for modelling both permission and exemption and can be iterated at any level of depth.

By using the T-Norm model we can specify different types of exceptions. The *first type* is represented by exceptions to norms activation. When some specific conditions on the event that activates the norm are met, the consequent deontic relation has not to be generated. An example of this type of exception to Norm01 is: “*ambulances do not have to pay for entering into the limited traffic area*”. In this case a check on the type of the vehicle inhibits the creation of the obligation to pay, thus creating an exemption.

The exceptions of the *second type* are those to deontic relations, i.e. when some specific conditions on the regulated event are satisfied the generation of violation/fulfilment is inhibited. An example of this type of exception to Norm02 is: “*school teachers can always borrow every DVD from the library*”. In this case, a check on the position of the borrower can be used to prevent the violation of the prohibition, that is, for creating a permission. This type of exception cannot be expressed by inhibiting the activation of the norm (using the first type of exception) because the condition (being a school teacher) is on the borrower, who is not part of the activating event of the prohibition.

2.4. The T-Norm model of norms

Both these types of exceptions could be expressed by adding some specific conditions to the antecedent of a norm. However, this solution requires to modify an already enforced norm by adding further conditions. This is not a good solution because it implies changing a previously defined norm every time an exception is introduced. A better solution consists in expressing exceptions with a construct that is external to norms. Our idea is to introduce a construct that is able to inhibit the activation or the fulfilment/violation of a norm when the activating or the regulated event happens and some further conditions are met. Therefore, we formalize exceptions with a construct whose effects is to inhibit the activation of one of the components of a norm. Given that an exception is strictly related to a norm, we assume that it has access to all the variables introduced in the related norm, to which it simply adds some more conditions. An exception is expressed in one of the following ways on the basis of its type:

```
EXCEPTION TO Norm_n TYPE 1
ON ?event1
WHERE conditions on event1
THEN exceptionToNorm(Norm_n,?event1)
```

```
EXCEPTION TO Norm_n TYPE 2
ON ?event2
WHERE conditions on event2 AND isGenerated(?dr, Norm_n)
THEN exceptionToDR(?dr,?event2)
```

For example the formalization of an exception of the fist type to Norm01 for ambulances is:

```
EXCEPTION TO Norm01
ON ?e1 WHERE Ambulance(?v)
THEN exceptionToNorm(Norm01,?e1)
```

These types of exceptions cannot be formalized by simply deleting a norm or a *general deontic relation*, because they suspend the effects of norms only in particular situations.

2.4. The T-Norm model of norms

For example, Norm01 applies to all vehicles except ambulances and Norm02 applies to all library subscribers except school teachers.

By analysing real cases of norms, we realized that there exists a *third type* of exceptions whose effect is to inhibit the fulfilment or violation of *specific deontic relations*. These exceptions are different from those of the second type because they are triggered by an event that is not the one regulated by the norm. For example an exception to the Covid-19 Norm03 is: “*if the house is on fire then everybody is allowed to leave it*”. This exception is activated by an event (the house is on fire) that is different from the action that is regulated by the norm (leaving the house). We model those exceptions in the following way:

```
EXCEPTION TO Norm_n TYPE 3
ON ?event_n
WHERE conditions on event_n AND isGenerated(?dr, Norm_n) AND
      NOT fulfills(?agent,?dr) AND NOT violates(?agent,?dr)
THEN exceptionToDR(?dr,?event_n)
```

For example the exception to Norm03 is formalized as⁷:

```
EXCEPTION TO Norm03 TYPE 3
ON ?en
WHERE Fire(?en) AND place(?en,?house) AND residence(?house,?agent) AND
      isGenerated(?dr, Norm03) AND activated(?dr,?e1) AND
      affectedPerson(?e1,?agent) AND NOT fulfills(?agent,?dr) AND
      NOT violates(?agent,?dr)
THEN exceptionToDR(?dr,?en)
```

It is also possible to have exceptions to exceptions that will inhibit the activation of the three types of exceptions described above.

⁷Where `affectedPerson` is the property that connects the event of having a positive swab with the tested person and it is used for connecting the activation event of the norm with the activation of the exception.

2.5 Operational Semantics of the model

In this section, we will show how the model of norms proposed so far can be used to monitor the temporal evolution of normative states on the basis of the events occurred in the interaction among agents. Our goal is to compute the violation or fulfilment of norms on the basis of actual events.

The operational semantics of the T-Norm model can be specified by providing an unambiguous procedure for translating the model into a target formalism that already has an operational semantics. As target formalism we choose production rules, because their structure and behavior make it fairly easy to translate norms into them. Production rules, often simply called productions or rules, have been investigated in computer science, and in particular in the AI literature related to knowledge representation and reasoning [32]. A production rule has the form:

IF *conditions* THEN *actions*.

It has two parts: an antecedent set of *conditions* that are tested on the current state of the *working memory* and a *consequent* set of *actions* that typically modify the *working memory*.

The operational semantics of a production rule system is given in the W3C Recommendation of the RIF Production Rule Dialect⁸ by means of a labeled terminal transition system. Such an operational semantics depends on the adoption of a *conflict resolution strategy* for selecting the rule instance that must fire when more than one rule is applicable. Our conflict resolution strategy is as follows. Firstly, use the *priority* among rules (for example, as we will discuss later, production rules for representing exceptions have higher priority than production rules for expressing norms). Secondly, when two or more rules have the same priority, use the *order* conflict resolution strategy, i.e., pick the first applicable rule in order of presentation. This choice will not influence the final state reached by the working memory because the actions of the production rules used for expressing norms will never remove knowledge from the *State KB*, they have effects only on the *Deontic KB*.

⁸https://www.w3.org/TR/rif-prd/#Operational_semantics_of_rules_and_rule_sets

2.5. Operational Semantics of the model

We will now describe the procedure for translating every norm written using the T-Norm model into three production rules and every exception into one production rule. In particular, every norm (Norm_n) translates into three production rules according to the following procedure:

1. Create one production rule equal to the first ON...THEN part in the norm. Add, among the *conditions* part of this production rule, the condition for managing exceptions of the first type (i.e. $\text{NOT exceptionToNorm}(\text{Norm}_n, ?e1)$), this condition is satisfied if in the working memory there is not an exception to Norm_n that matches with the activation event.
2. Create one production rule equal to the second ON...THEN part in the norm. Add, among the *conditions* part of this production rule, the condition for managing the exception of the second and third type
(i.e. $\text{NOT exceptionToDR}(?dr, ?e2) \text{ AND } \text{NOT exceptionToDR}(?dr, ?en)$), the conditions for checking that the regulated action is performed before event3 (that for obligations may represent a deadline) and after the activation of the norm, and the conditions for checking that the deontic relation, which can be matched with the rule, is generated by Norm_n and it is not already fulfilled or violated.
3. Create one production rule for expressing the ON...ELSE part in the norm. This rule is fired when the regulated action (represented in the norm with variable event2) can no longer be performed before the event represented with the variable event3 . That is, when event3 has occurred (e.g. the deadline has passed) and the regulated action (e.g. the payment) has not been executed. The procedure adds, among the *conditions* of this production rule, the condition for checking that event3 is happened and that the deontic relation, generated by Norm_n , is not already fulfilled nor violated. As for the previous rule, the procedure adds also the conditions for managing the exception of the second and third type.

Every exception to a given norm (Norm_n) translates into one production rule thanks to another automatic procedure. Since each exception has access to all variables introduced in the related norm, the conditions in the norm are merged with the conditions of the ex-

2.5. Operational Semantics of the model

ception during the creation of the production rule. In particular, conditions on *e1* (asserted in the corresponding norm) are added to the conditions of the production rule created from exception of the first type. Conditions on *e2* (asserted in the corresponding norm) are added to the conditions of the production rule created from exception of the second type. Production rules that represent first-, second-, and third-type exceptions must fire before production rules that express norms, so they have a higher *priority* than the latter. Since production rules, used to formalise exceptions to norms, act before the production rules of the norms themselves, they are able to inhibit the norm for certain events.

Exceptions to exceptions are able to inhibit an exception to norm for certain events. They must fire firstly, thus production rules for representing *exceptions to exceptions* must have higher priority than the production rules for expressing exceptions to norms. In order for one exception to exception to inhibit the activation of one specific exception to Norm_n, it is required to add to the production rule of the latter a further condition for checking that does not exist an exception to exception for its activation event, i.e. NOT `exceptionToException(Normn`

The conditions of the production rules are evaluated on a *working memory*, which consists of: (i) the *State KB* where all the relevant events happened and the actions performed by the agents are recorded, those events are represented using the *OWL Event Ontology*; and (ii) the *Deontic KB*, where all the information for managing the evolution of the state of norms is stored.

Given that the *working memory* contains an OWL ontology, it is possible to use OWL reasoning on its content for computing for example that the performance of an action implies another one. This is a crucial aspect of the proposed normative model because without any further addition, it is possible to reason on the effects that the logical implication between actions has on norms fulfilment or violation. In fact, we obtain that the obligation to perform an action is fulfilled by any action that implies the regulated one. This is because we have the following chain of implications: action *a1* implies action *a2* and *a2* produces a fulfilment, so the performance of *a1* leads to a fulfilment. For example, since selling an object to someone involves a transfer of ownership, a sale will fulfill the obligation to transfer ownership of an object to someone. Similarly, the prohibition to perform an action

is violated by the performance of any action that implies the regulated one. For example, since the reproduction of an audio file implies its use, if the use of a particular audio file is prohibited its reproduction will lead to an violation. Finally, the permission to perform a generic action implies the permission to perform all the more specific actions implied by the generic one. This is because the specific action implies the more generic one that will activate the exception that in turn inhibits the norm. For example, if an agent has permission to transfer the ownership of a product, through OWL reasoning it is possible to infer that she also has permission to sell or give someone else the product.

2.6 Architecture of the framework and its implementation

The architecture of the framework designed to compute the fulfillment or violation of a set of norms (formalized with the T-Norm model) is depicted in Figure 1. In the proposed framework, we take advantage of two types of computation: OWL reasoning on the *State KB* and forward chaining realized by means of production rules. OWL reasoning and forward chaining are combined in a safe manner because they alternate. In particular, the main steps of a software able to simulate the evolution of the norms state over time, is as follows:

1. Every time an event or an action occurs its representation is added to the *State KB*, then an OWL reasoner is executed on the working memory. We assume that only events that happen at the current instant of time can be inserted in the *State KB*;
2. Then run the forward chaining engine on the working memory resulting from the previous step using the production rules generated from the norms and from the exceptions and store the resulting *State KB* together with the *Deontic KB* in the working memory;

2.6. Architecture of the framework and its implementation

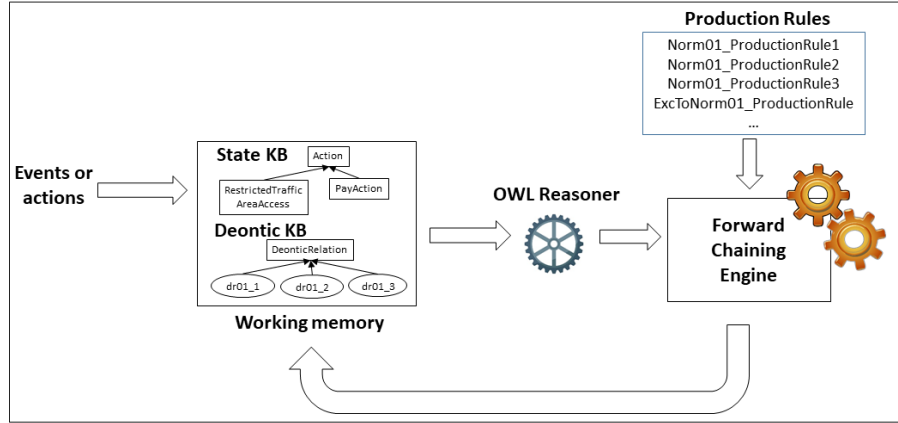


Fig. 2.1: Architecture of the framework designed to compute the fulfillment or violation of norms.

3. Updates the variable that keeps track of the current instant of time to the next *significant* time instant⁹ and go back to point 1.

In our model we need to combine OWL reasoning and forward chaining because it is not possible to use only OWL reasoning for computing the violation or fulfillment of norms. In fact, when norms regulate time constrained actions, it is necessary to deduce that the non-performance of the regulated action before a deadline implies violation or a fulfillment. Given that OWL reasoning works on an open world assumption, inferences of this type cannot be drawn directly. One possible solution to this problem is computing the closure of specific classes using an external routine as proposed in [58]. The advantage of using production rules is a clear separation of two different types of computation, each one used coherently with its nature, and having a more declarative solution where the semantics of norms is expressed with production rules instead of using Java code.

We tested the described framework by implementing a Java prototype that uses Pellet¹⁰, an open-source Java based OWL reasoner, and the JENA general purpose rule engine¹¹ for realizing forward chaining on production rules. The reason why we chose to use the JENA framework is that, differently from other rule-based systems like DROOLS (used in

⁹An instant of time is significant when its occurrence is significant for at least one norm, e.g. it is the time instant in which a deadline expires or the time instant at which an event or an action occurs.

¹⁰<https://github.com/stardog-union/pellet>

¹¹<https://jena.apache.org/documentation/inference/#rules>

2.6. Architecture of the framework and its implementation

[11]) or Jess (used in [68]), its rule engine natively supports rule-based computations over an OWL ontology serialized as an RDF graph. JENA provides forward chaining realized by means of an internal RETE-based interpreter.

To test the framework, the various type of norms discussed in Section 2.4.1 were manually translated into a set of production rules written using rule syntax and structure of the JENA rule-based reasoner. The translation is done following the procedure described in Section 2.5. Given that the JENA rule engine does not natively support the possibility to specify the *priority* among rules, we introduced a variable called `salience` whose value change from 0 to 2 and a new builtin called `isSalience(n)` that can be used in production rules for checking if the value of the `salience` variable is equal to `n`.

In order to simulate the evolution in time of the state of the norms, a set of real actions matching with the activation condition of the norms or with their regulated actions have been inserted in the *State KB*. As depicted in Figure 2, in order to check the fulfillment or violation of the different deontic relations created by the various activation of Norm01, we entered three accesses to the restricted traffic zone in the *State KB*. For one of these accesses (`access3`) we have entered the corresponding payment and therefore the deontic relation that obliges the owner of the vehicle to pay becomes fulfilled. For another access (`access2`) we do not enter the corresponding payment and thus, when the deadline expires, the deontic relation that expressed the obligation to pay becomes violated. For the last of the accesses (`access1`), the vehicle is an ambulance, so thanks to the exception, the obligation to pay is not even created. The Java project developed for simulating the fulfillment or the violation of Norm01, the three production rules generated starting from Norm01 and the production rule generated from the exception to Norm01 (see Section 2.4.2) are available on GitHub¹².

¹²<https://github.com/fornaran/tnorm.Norm01>

2.7. Related Work

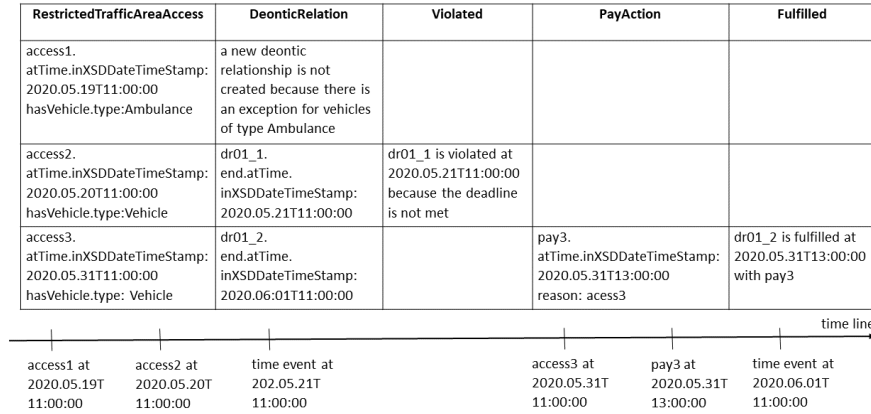


Fig. 2.2: Fulfillment/violation of the different deontic relations created by the activations of Norm01.

2.7 Related Work

In the literature, there are various proposals where models of norms and policies are formalized using different languages and where different frameworks are investigated with the goal of providing various services. Useful services are: searching of policies having certain characteristics [120], anticipating conflicts among policies [145], monitoring [60] or compliance checking [44], and simulation for performing a what-if reasoning [157].

One of the pioneer techniques for normative reasoning is deontic logic [161]. Despite deontic logic approaches present some limitations, for example the triggering and regulated actions are usually expressed with simple propositional formulae [75], some of their basic concepts and insights are still used in many recent approaches where other formal languages are used. In order to pursue interoperability among different normative systems, it is crucial to use a standard language for the formalization of norms. Today's there are two standards: the previously mentioned ODRL policy expression language, which is a W3C Recommendation and the OASIS standard LegalRuleML¹³, which defines a rule interchange language for the legal domain and is formalized using RuleML. ODRL has many connections with the model proposed in this paper as it is a language for expressing obligations, prohibitions, and permissions. A great limitation of ODRL is not having an

¹³<https://www.oasis-open.org/committees/legalruleml/>

operational semantics that allows to compute the fulfillment or the violation of policies. In our previous work [60] we proposed to extend the ODRL information model to express its operational semantics using finite state machines implemented using production rules. In this work we have moved further away from ODRL, in order to overcome some limitations. Firstly, in ODRL it is not possible to specify generic policies applicable every time to a different agent. In ODRL the debtor of a policy can only be a specific agent. Differently, thanks to our abstract model for policies specification, it is possible to apply one policy to all the agents who will perform a certain action (for example having a positive swab) or who plays a certain role. Secondly, we do not consider exceptions (and in particular permissions) at the same level of obligations and prohibitions. From our perspective exceptions are derived concepts and they exist only if there is a corresponding basic level construct that expresses obligations and prohibitions. Finally, while ODRL has a fixed set of deontic types, in our model we focus on specifying the sequence of events that bring to a violation or to a fulfillment. An important aspect that the T-Norm and the ODRL model have in common is the use of semantic web technologies. Although they use them for different purposes: ODRL uses the OWL language for the specification of the policy meta-model, while in the T-Norm model, as well as in the OWL-POLAR model [145], the OWL language is used for modeling the actions performed by agents and consequently to express the activation conditions and the actions regulated by the norms.

We now continue our comparison by focusing on models of norms that are expressed using semantic web languages and/or by using production rules, although none of them combines OWL reasoning and productions as we do. Two features that make our model innovative are: the formalization of the relation between norms and time constraints and the possibility to directly describe what sequence of events or actions would bring an agent to the violation or to the fulfilment of a norm. In the OWL-POLAR framework [145], similarly to our approach, the state of the world is represented using an OWL ontology. Differently, policies activation is computed by translating the conjunctive semantic formulae, used for describing what is prohibited, permitted or required by the policy, into SPARQL queries that are evaluated on the state of the world. In this work we propose a straighter approach where norms conditions are directly evaluated on the state of the world without the need

2.7. Related Work

of translations. An interesting aspect of the OWL-POLAR framework that we plan to investigate in our future works, is the mechanism for anticipating possible conflicts among policies, and for conflict avoidance and resolution.

In [11] one type of norm is defined as a tuple that can generate a norm instance that in turn can be fulfilled or violated. A norm specifies a target condition that describes the state that fulfills the norm and a maintenance condition used for defining the conditions that, when they no longer subsist, lead to a violation. In this approach count-as rules are used to introduce institutional facts, regulated by norms, starting from brute events. Differently from our model, where it is possible to model different types of deadlines, in this approach, only a timeout property, i.e. a deadline for the reparation of the violation of a norm, is taken into account. In [11] an interesting violation handling norm is formalized that is activated when another norm is violated. Similarly to our approach, the monitoring is realized using a production system that concretely is implemented using DROOLS, but no discussion is offered on the advantages of using OWL reasoning and on how to combine it with forward-chaining realized by means of production rules.

Another interesting proposal is the KAoS policy management framework [157]. In KAoS Semantic Web technologies are used for policy specification and management, in particular, policy monitoring and enforcement is realized by a component that compiles OWL policies into an efficient format. In the literature there are other interesting approaches where norms are specified as rules but they are not taking advantage of the use of semantic web technologies. For example in [158] norms are generators of commitments for the agents playing a certain role in an artificial institution. In [68] norms have a type, they may have a deadline, and given that their form is: *preconditions* $\rightarrow$ *postconditions*, those norms are easily expressible with Jess rules¹⁴. Finally in [44] an extension of the ODRL language is proposed to capture the semantics of business policies thanks to their translation into Answer Set Programming for making it possible to realize compliance checking. An interesting aspect of this work is that the result of compliance checking can be positive or negative with an explanation of the aspects of the policy that caused the non-compliance.

¹⁴Jess is a rule engine for the Java platform

2.7. Related Work

In our future work, we plan to investigate the application of sanctions or rewards and to study the formalization of the notion of institutional power, and we plan to further investigate the expressive power of the model for specifying other types of deontic relations.

3

A Methodology for Formalizing Different Types of Norms

3.1	Abstract	64
3.2	Introduction	64
3.3	Related works	66
3.4	The T-Norm model	68
3.5	Methodology	70
3.5.1	Using ontologies for modeling norms	71
3.5.2	Norms with activation condition	74
3.5.3	Representing obligations and prohibitions	77
3.5.4	Temporal aspect of norms	78
3.5.5	Specific and General Deontic Relations	80
3.6	Conclusions	81

Published Manuscript: Roshankish, S., Fornara, N. (2022). A Methodology for Formalizing Different Types of Norms. In: Baumeister, D., Rothe, J. (eds) Multi-Agent Systems. EUMAS 2022. Lecture Notes in Computer Science(), vol 13442. Springer, Cham. <https://doi.org/10.1007/978-3-031-20614-620>

3.1 Abstract

In a world where many activities are carried out digitally, it is increasingly urgent to be able to formally represent the rules, norms, and policies that regulate these activities. In multi-agent systems, formalizing policies written in a natural language into a formal model, making them machine-readable, is a demanding task. In this paper, we introduce a methodology to help people to understand the fundamental elements that they should consider for this transformation. In this paper we will focus mainly on a methodology for formalizing norms using the T-Norm norm model, this because it allows us to express a rich set of different types of norms. In any case, the proposed methodology is general enough to also be used, in some of its steps, to formalize norms using other formal languages. This is an important issue because since there is not yet a set of different types of norms that is sufficiently expressive and is recognized as valid by the NorMAS community, papers presenting a given model usually do not explicitly state which types of norms can be expressed with that model and which cannot. Therefore, the second goal of this paper is to propose and discuss a rich set of norm types that could be used to study the expressive power of different formal models of norms and to compare them.

3.2 Introduction

In a world where many activities are carried out digitally, it is increasingly urgent to be able to formally represent the rules, norms, and policies that regulate these activities. By doing that, it is important to take into account that they are carried out by autonomous subjects who are able to decide to violate these rules. The activities regulated by the norms can be of various types and also include the very important ones related to the use, exchange and manipulation of the enormous amount of digital data that exist nowadays. Since these norms and policies can be violated (for example, it is very difficult to regiment obligations [56]) it is also urgent to propose mechanisms to automatically monitor compliance with these norms.

In the academic literature, there are an interesting number of general models of norms and policies. Some of these are close to being able to be used in real applications in today's Web as they are expressed with standard Semantic Web Technologies, which is a crucial characteristics for realizing interoperable systems. One of them is the W3C Recommendation ODRL 2.2¹ (the Open Digital Rights Language), which is a policy expression language that provides an information model and a vocabulary for specifying permissions, prohibitions, and obligations about the usage of digital assets and services. Two others

¹<https://www.w3.org/TR/odrl-model/>

3.2. Introduction

are the T-Norm model [62] and the OWL-POLAR model [145]. They are two semantic web based complementary models having an operational semantics for reasoning about norms and policies fulfillments and violations.

The papers that describe these models are mainly focused on the presentation of the model that is exemplified usually with the formalization of some examples of norms, regardless of their type. What is missing, however, is a *methodology* that explains what steps should be followed if one wants to start from a norm written in a natural language (e.g., English) and be able to choose the model for its formalization and use it to arrive at the formal specification of the norm, which can then be used to reason about it and verify its fulfillment or violation. Since there is not a commonly accepted set of *types of norms* in the literature, papers presenting a given model usually do not explicitly state which types of norms can be expressed with that model and which cannot. Thus in this paper we have the following two goals.

Our first goal is to explain the methodology that can be used by people to translate norms written in one natural language into a language specifically designed for the formal specification of norms. The proposed methodology consists in: first understand if the norm can be expressed with a certain model, that means to understand which *type* the norm belongs to and if the type is supported by the model; secondly come to a proper formalization of the norm using the chosen model, this by applying the methodology proposed in this paper. We will focus mainly on formalizing norms using the T-Norm norm model, this because it allows us to express a rich set of different types of norms. However, it is important to emphasize that the proposed methodology is general enough to also be used, in some of its steps, to formalize norms using other formal languages that have some similarities with the T-Norm model, such as at least OWL-POLAR and ODRL.

There are many reasons why it is interesting to specify norms using formal models. First, because norms become machine-readable, therefore it is possible to automatically analyse and query them like for example it is discussed in [120] where the PrivOnto ontology is used for analysing 115 privacy policies. For example, it will be possible to search the set of resources on which it is possible to perform certain actions based on the customers' interests. When a policy is formalized with a machine-readable formal language that has an operational semantics, it is also possible to provide services for compliance checking of policies [145, 125, 60, 62]. This functionality plays an important role especially in domains in which the customers' sensitive data is collected and companies need to monitor the compliance of customers' privacy. This functionality is important also to create a trustworthy environment for customers by providing monitoring platform that they can use to see whether their privacy policies (norms) are violated or not. For instance, a customer can attach to one picture the prohibition to publish it on a public platform for advertisement and would like to monitor if the actions which are performed on the picture are compliant with this prohibition.

3.3. Related works

Another reason why it can be useful to specify norms with formal languages is that it becomes easier for humans to understand their actual meaning, which is not always as immediate as it should be. For example, during the Covid-19 pandemic, it was not always easy to immediately understand what norms are in effect at any given time in a specific location and whether they entail obligations or prohibitions to perform actions. Another example of norms whose meaning and implications are not always immediately clear to the reader are the various privacy policies that regulate the processing of our data when we browse websites and use social networks. Users often accept such policies in order to use online services often without fully understanding what they mean, this is because they are too long or complex.

The second goal of this paper is to propose and discuss a rich set of norm types that could be used to study the expressive power of different formal models of norms and to compare them. Knowing that a certain model of norms is or is not capable of expressing a certain type of norms is fundamental to deciding which model to adopt in a certain application context. For example, if a norm generates a specific obligation that has a deadline, it is necessary to choose a model of norms that allows to express this temporal constraint and to verify its fulfillment. Secondly, once it is clear that a certain type of norm can be expressed in both language A and language B, it will also be possible to translate norms written with the first language into norms written with the second. Thus making systems that use different norm models interoperable, a fundamental aspect in today's world where one software agent must be able to interact with multiple open interaction systems without having to be reprogrammed every time.

This paper is organized as follows: in Section 3.3 the most relevant and recent papers presenting a model for norms and policies specification in which semantic web technologies have been used are discussed. In Section 3.4 the T-Norm model is briefly presented. In Section 3.5 the methodology proposed in this paper is described and the set of different types of norms is discussed. Finally in Section 3.6 we draw some conclusions.

3.3 Related works

In the multiagent systems community, over the last twenty years, many models of norms and policies for regulating the behaviour of autonomous agents has been proposed [12, 37]. In some of these models semantic web technologies have been used for modeling some components of norms/policies that can be used for expressing obligations, prohibitions, and permissions. The first proposals were the KAoS framework [33], the REI [93] policy language, and the PProvisional TrUst NEgotiation framework Pro-tune [29]. Those approaches are summarized and compared in [30] where the requirements for a policy framework are discussed and the various approaches are categorized discussing whether the policies are

3.3. Related works

public or not. For example, for the public policies, it is possible to use KAOS and REI frameworks as we need just one step evaluation to see if two policies are compatible. On the other hand, if a policy contains sensitive data, they are required to have *stateful and stateless negotiation* protocols for further security concerns.

An important policy language based on semantic web technologies, which is a W3C Recommendation since 15 February 2018, is the Open Digital Rights Language (ODRL 2.2). It is a policy expression language that can be used to represent permitted, prohibited, and obliged actions over a certain asset. ODRL policies may be limited by constraints (e.g., temporal or spatial constraints). ODRL was originally (in 2001) an XML language for expressing digital rights, that is, digital content usage terms and conditions. In version 2.0 and 2.1 ODRL is a Policy Language formalized in RDF with an abstract model specified by an ontology. It has no formal semantics, so compliance checking of policies written with this language cannot be performed automatically. An interesting attempt to give a formal semantics to ODRL 2.1 policies is presented in [149]. Some extensions of ODRL have been proposed to overcome some of its limits. In particular, in [60] an extension of the ODRL Information Model has been proposed together with a set of state machines used for describing the evolution in time of the deontic state of obligations, prohibitions, and permissions. Another extension of ODRL is presented in [44] to model both regulatory policies (in the form of nested permissions, prohibitions, obligations and dispensations), and business policies via discrete permissions. A policy written with that extension of the ODRL language is then translated into an Institutional Action Language (InstAL) policy and thanks to its formal semantics, expressed in Answer Set Programming, it is possible to automatically check compliance and also provide an explanation of the aspects of the policy that brings to the non-compliance. In [72] a specific use case drawn from the social networks field is used to validate the expressiveness of the ODRL 2.0 model.

Other two interesting proposals of a policy/norm model and framework, which are based on semantic web technologies, are OWL-POLAR [145] and T-Norm [62, 63] models. Those policies/norms models and their expressivity will be discussed in Section 3.4. An interesting aspect that differentiates the two models is the way in which the two models define mechanisms to reason about policies to test whether agents' behavior satisfies them or not. In the OWL-POLAR a query answering mechanism (DL-safe) has been used to check if any action satisfies the policies. In the T-Norm model a rule-based approach is used that brings the generation of several deontic relations used to represent obligations and prohibitions generated by the activations of norms. In addition, the T-Norm model makes it possible to formalize the temporal constraints that exist between the activation of a norm and the class of actions regulated by the norm.

Other interesting models of norms that, like the T-Norm model, are rule-based are: the one proposed

3.4. The T-Norm model

by Garcia-Camino et al. [68] where rules are operationalized using the JESS a rule engine for the Java platform; and the one proposed in [11] where reasoning on norms is realized with DROOLS a business rule management system. Another interesting proposal is the OASIS standard LegalRuleML², which defines a rule interchange language for the legal domain and is formalized using RuleML.

Logic and Knowledge Engineering Framework and Methodology (LOGIKEY) is another interesting framework which was introduced recently in [22]. The main objective of this framework is to enable and support the practical development of computational tools for normative reasoning based on formal methods. In their approach, they use higher-order logic (HOL) as the formal framework. They also used some GDPR examples to show how their framework supports the ethical and/or legal (echico-legal) domains theories.

3.4 The T-Norm model

The *T-Norm* model can be used to formalize a precise and rich set of *types of norms* that regulate the interactions between autonomous agents. Namely (as we will further discuss in the paper): (i) norms with a activation condition represented by a class of events; (ii) norms without an activation condition; norms that generate (iii) general or (iv) specific obligations or prohibitions to perform (or not to perform) actions that can be constrained to happen before something else happens, (v) exceptions to those norms; (vi) exceptions to obligations and prohibitions (i.e. exemptions and permissions respectively). Once a set of norms is formalized, a specific *framework* has been proposed to automatically check if the agents' behavior conforms or does not conform to the given set of norms. This is done by simulating or monitoring the evolution of the state of those set of norms as time passes, events occur and autonomous agents perform actions. The framework for norms monitoring has been proposed by taking into account the *operational semantics* of the T-Norm model. The model, its operational semantics, and the framework were introduced in [62, 63].

The *T-Norm* model captures the following intuitive meaning of norms: whenever a particular *activation condition* is satisfied (i.e. an event that belongs to a particular class of events occurs) a *deontic relation* (general or specific) is created for regulating the performance of a class of actions by certain agents. In turn, every time an action belonging to the *class of the regulated actions* is executed before a certain event happens (for example a certain temporal event representing a deadline) and the deontic relationship represents an obligation it will be *fulfilled*, while if the deontic relation represents a prohibition it will be *violated*. On the contrary, when an action belonging to the class of regulated actions can no longer

²<https://www.oasis-open.org/committees/legalruleml/>

3.4. The T-Norm model

be performed (for example because the deadline has expired) and the deontic relationship represents an obligation it will be *violated*, while if the deontic relation represents a prohibition it will be *fulfilled*.

In order to formally describe such a dynamic behaviour, the abstract model of a norm cannot consist only of a set of facts (like it is in many models of norms and policies, e.g. ODRL³, OWL-POLAR [145], and the model proposed in [11]). In all these models the intrinsically dynamic nature of norms is described in their semantics or is left to their intuitive meaning described in the text. The T-Norm model allows to specify how the performance of certain actions or the occurrence of certain events will change the state of the interaction among agents. Therefore the basic building blocks of the T-Norm abstract norm are *rules* of the form ON . . . THEN . . . ELSE⁴. The abstract norm has not a pre-defined deontic type, as discussed in Section 3.5, it is those who formally specify a norm who will decide whether the norm activation creates obligations or prohibitions. In the T-Norm model a generic abstract norm has the following form:

```
-----  
Abstract Norm  
-----  
1:  NORM Norm_n  
2:  [ON ?event1 WHERE conditions on ?event1  
3:    THEN  
4:      COMPUTE]  
5:    CREATE DeonticRelation(?dr);  
6:    ASSERT isGenerated(?dr, Norm_n); [activated(?dr, ?event1);]  
7:    ON ?event2 [BEFORE ?event3 WHERE conditions on ?event3  
8:      WHERE actor(?event2, ?agent) AND conditions on ?event2  
9:    THEN ASSERT fulfills(?agent, ?dr); fulfilled(?dr, ?event2) |  
10:      violates(?agent, ?dr); violated(?dr, ?event2)  
11:  [ELSE ASSERT violates(?agent, ?dr); violated(?dr, ?event3) |  
12:    fulfills(?agent, ?dr); fulfilled(?dr, ?event3]
```

In the proposed model the first (optional) ON . . . THEN component (line 2,3) is used for expressing those norms that have an activation condition. The second ON . . . THEN component (line 7,9) is used for expressing that when a specific action, which belongs to the

³<https://www.w3.org/TR/odrl-model/>

⁴The ON clause has been chosen instead of the more common IF clause to highlight that the part after THEN is executed when a particular event or action occurs and not simply when a condition is satisfied

class of actions regulated by the norm, is performed (before something else occurs) there will be a fulfillment or a violation. In alternative, the ELSE part of the second rule (line 11) will be followed when an action that belongs to the class of the regulated actions can no longer be performed.

The *formulas* used in the abstract norm are conjunctions (in the WHERE part) or sequences (in the CREATE and ASSERT part) of *atomic assertions* written using the classes (unary predicates starting with capital letter) and the properties (binary predicates starting with a lowercase letter) defined in the *T-Norm Ontology* depicted in Figure 1⁵. Variables (starting with '?') refers to individuals. Variables used in the WHERE parts of the norm for expressing conditions on events can be used freely and have to be bound to individuals in the *State Knowledge Base* (where the interaction among agents is represented) for the conditions to be met. In the WHERE parts it is also possible to compare the value of a variable with a constant value using any of the symbols {<, >, =, ≠, ≤, ≥}. A constant is a numerical value or an individual in the ontology. Variables that appear in the ASSERT part of a norm must have been introduced previously in one of its ON or CREATE parts. In the COMPUTE part some values can be calculated (for example the deadlines) using the value of previously introduced variables⁶.

3.5 Methodology

In this section, we describe the various steps of the procedure to be followed to transform a norm written in a natural language (for example in English) into a norm written using a formal machine-readable language like the *T-Norm* model. As it will be discussed, some steps of the described procedure can also be used to formalize norms using the OWL-POLAR or the ORDL normative language. Starting from a norm expressed in natural

⁵The T-Norm ontology in OWL is available at <https://raw.githubusercontent.com/fornaran/T-Norm-Model/main/tnorm.owl>

⁶The choice of using conjunctions or sequences of atomic assertions (analogously to what is proposed in OWL-POLAR to express the various components of their norm model) has the advantage of avoiding requiring the user of the model to learn a specific formal language, once written those expressions can be easily and automatically translated into the conditions or actions of production rules or into SPARQL queries)

3.5. Methodology

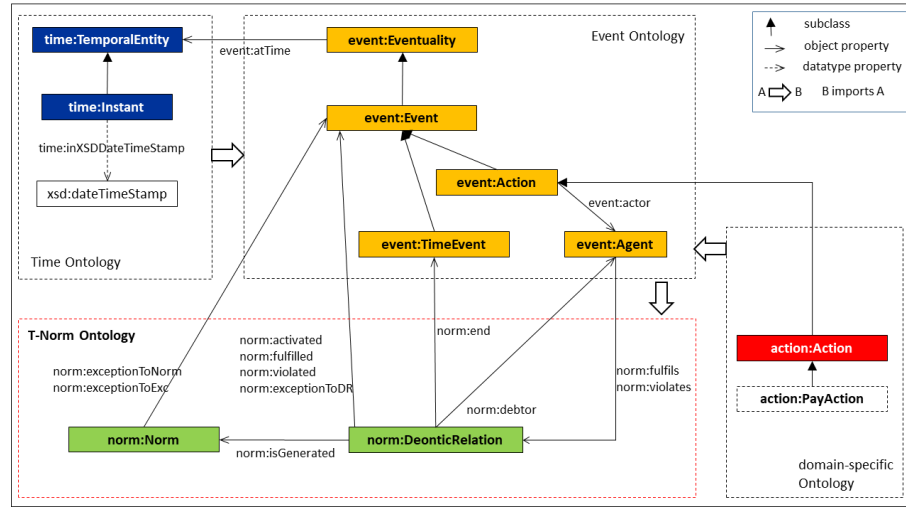


Fig. 3.1: The T-NORM Ontology and its connections with other ontologies.

language, following each step of the methodology, the *Abstract Norm*, introduced in the previous section, is made more concrete to the point of being the formalization of the norm from which the process started. In the following, we use two real running examples. We call the first example Norm1, which is inspired from the law regarding the access to limited traffic area in Milan city such that "when an agent enters in the limited-traffic area of Milan, between 7:30 and 19:30, they have to pay 5 euros before 24:00 on the day of entry"⁷. The second example, called Norm2, is the rule that must be followed by libraries in Italy regarding lending of DVDs, it is "Italian libraries cannot lend DVDs until 2 years are passed from the distribution of the DVD"⁸.

3.5.1 Using ontologies for modeling norms

In the first step of the procedure, we need to define (or search among the existing ones) one or more formal ontologies to represent the classes of events or actions that are relevant for the norm that we want to formalize. As it is discussed in the previous section, for every norm normally three *classes of events/actions* should be specified:

⁷<https://www.comune.milano.it/aree-tematiche/mobilita/area-c>

⁸According to Art. 69 c.1 of the Copyright Law (22-4-1941, no. 633)

3.5. Methodology

- The class of events that represent the activation condition of the norm (described using ?event1 in line 2 in the *Abstract Norm*);
- The class of actions regulated by the norm (described using ?event2 in line 7,8 in the *Abstract Norm*);
- The class of events defined for constraining the performance of the actions regulated by the norm. One action, belonging to the class of the regulated ones, should or should not occur before an event belonging to this constraining class (described using ?event3 in line 7 in the *Abstract Norm*).

Those classes are specified in the WHERE parts of the norm and are represented using the classes and properties defined in formal ontologies. In the *T-Norm* model and in *OWL-POLAR* model, the W3C Web Ontology Language (OWL 2) is used for specifying ontologies. OWL is a Semantic Web language designed to represent knowledge about things, groups of things, and relations between things. An important advantage of using OWL is that it is a well-known standard language, which can make it easier for those who want to formalize their norms. Moreover, the formal semantics of OWL makes it possible to perform automatic reasoning on the state of the world, an operation that has important consequences on the computation of the *deontic force* (it is obliged or it is prohibited) associated to the actions performed by the agents. In ODRL the information model of the language is formalized using an OWL ontology, while the actions, the parties, and the assets involved in one ODRL policy are described using the ODRL Vocabulary⁹.

It is possible to use several different ontologies for representing class of actions and their properties inside one T-Norm norm or one OWL-POLAR policy. For compatibility reasons, we suggest to use ontologies that are compatible with OWL ontologies, this because the chosen ontologies (each one referred to as *domain-specific Ontology* in Figure 1) should be imported into the *Event Ontology* which is written by using the OWL language.

We will now exemplify the formalization of two classes of events necessary for the formalization of Norm1. Norm1 is activated every time a vehicle enters the restricted traffic zone

⁹<https://www.w3.org/TR/odrl-vocab/>

3.5. Methodology

of the city of Milan. We assume that `RestrictedTrafficAreaAccess` is a class of actions, `vehicle` and `owner` are two properties having as domain the `RestrictedTrafficAreaAccess` class, which are defined in an OWL domain-specific Ontology. We can then specify the class of events that activates Norm1 as (where `?event1` is shortened to `?e1`):

```
ON ?e1 WHERE RestrictedTrafficAreaAccess(?e1) AND vehicle(?e1,?v) AND  
owner(?v,?agent) AND atTime(?e1,?inst1) AND inXSDDateTimeStamp(?inst1,?t1)  
AND ?t1.time()>07:30:00 AND ?t1.time()<19:30:00
```

In the previous expression the variable `?agent` is introduced because it will be used in the second part of the norm to recognize who fulfills or violates the norm. After representing the activation condition of the norm, we need to formalize the class of actions regulated by the norm. For Norm1, the class of actions is the payment of 5 euros before 24:00 on the day of entry. For formalizing it we can for example use the `PayAction` class defined in the *Schema.org* vocabulary, which has an OWL version. *Schema.org* provides a collection of types and properties available at the URL `schema.org`. The major search engine providers use the *Schema.org* markup to improve the searching and the display of search results. This vocabulary has been designed by and is controlled by these organizations and represents an interesting attempt to realize a lightweight ontology that can be reused in different applications.

As mentioned earlier, the class of events described with the variable `?event3` has the role of constraining the time interval in which the action belonging to the class of actions regulated by the norm (`?event2`) shall or cannot be performed. In Norm1, the time interval when the payment action should be performed is constrained by a deadline (referred with the variable `?paymentDeadline`), i.e. the payment action must occur before 24:00 on the day of entry into the limited traffic zone. The formalization of norms where `?event3` is a time event are discussed in Section 3.5.4. To be able to express facts about the topological relationships (ordering) between instants and intervals, along with information about their length, and their value in terms of dates and times, we imported the *W3C Time Ontology*¹⁰

¹⁰<https://www.w3.org/TR/owl-time/>

into our *Event Ontology* both written in OWL. However, in the formalization of other norms the time constraint could be expressed with any class of actions (e.g. the payment must be made before leaving the restricted area) and in this case we will need to use another ontology to represent that class. The class of actions regulated by Norm1 can be specified as (where ?e1 and ?agent are variables introduced in the previous ON clause):

```
ON ?e2 BEFORE ?paymentDeadline
  WHERE PayAction(?e2) AND reason(?e2,?e1) AND recipient(?e2,Milan)
  AND price(?e2,5) AND priceCurrency(?e2,euro) AND actor(?e2,?agent)
```

3.5.2 Norms with activation condition

The goal of this section is to explain how to recognize whether a norm has an *activation condition* or not. In every norms or policy models the activation condition may describe a *class of events/actions* or as a *state of affairs*.

In the T-Norm model, the activation condition is the description of a particular class of events or actions. When an event that belongs to the activation condition class occurs, a new deontic relation is created and some temporal parameters may be computed. In order to recognize the activation condition in the text of a norm, we have to look for the events or actions that induce the model to activate obligations or prohibitions. The temporal relation between an event or action that satisfy the activation condition and the action that should or should not be performed is crucial: the activation condition must be satisfied before the obligation or the prohibition to perform a certain class of actions starts to hold. The instant of time at which the activation condition of a norm is satisfied by an event or action is very important because it can be used to calculate the deadline of obligations generated by the norm or the instant of time at which a prohibition ceases to subsist. For example in Norm1, the activation condition is represented by the class of actions regarding entering the Milan limited traffic zone and it is used for computing the deadline for the payment. In Norm2, the activation condition is given by the class of actions with which a DVD distribution is initiated. It is important to note that in the T-Norm model the activation condition cannot

3.5. Methodology

describe *a state of affairs*, although often a state of affairs is the result of an event and therefore the description of the class of events may substitute the description of the state of affairs as activation condition.

The reason why, in a norm model, a class of events and a state of affairs are treated differently is mainly due to their ontological difference: an event when it has happened it can no longer be retracted, a state of affairs can be satisfied at a certain instant of time and it can become unsatisfied subsequently. This is a crucial difference, because in the T-Norm model any satisfaction of the activation condition leads to the permanent creation of deontic relations. This permanent creation is important when the deontic relation regulates a class of actions that should or should not be performed in an interval of time and when the deontic relation itself can generate many violations and fulfillments as it is discussed below when Norm2 will be formalized.

The OWL-POLAR model for policies can be used when the activation condition α describes a state of affairs, like for example “a person is obliged to leave a location when there is a fire risk” or “when a person has a child which is under 18 they have to pay their tuition”[145]. In the OWL-POLAR model a policy is activated for a specific agent when the world state is such that the activation condition holds for that agent and the expiration condition does not hold. Therefore, at the time of activation it is necessary to know the specific agent for whom the policy is being activated, and this is not the case when the activation of a policy leads to the creation of *general deontic relations*, as will be discussed below. In the OWL-POLAR model the time constraint that exists between the initial satisfaction of the activation and the subsequent activation of the policy is not explicitly represented in the norms model, it is expressed in the description of how it is possible to reason about policies, therefore it is not in the model but inside the algorithm proposed for reasoning on policies.

There is another important distinction between an activation condition that describes a class of events and the one that describes a state of affairs. When it describe a state of affairs, it may make sense to ask whether the condition in the text should be formalized as an activation condition or as a set of conditions that restrict the class of actions regulated

3.5. Methodology

by the policy. For example in [142] *conditional norms* are discussed and the following example of a norm with a condition is described as: “it is prohibited to litter as long as there is a rubbish bin within x meters from an agent”. The condition of being within x meters from a rubbish bin may be modelled as an activation condition in some cases, and it can be considered as a condition that constrains the class of actions regulated by the policy in other cases. In this second case the policy can be modelled with the T-Norm model, and it is a prohibition (without activation condition) to perform the following class of actions: littering when the actor of the action is within x metres from the rubbish bin. If one action belonging to that class is performed then there is a violation of the prohibition.

The choice between the first and the second formalization depends on the type of reasoning that the norm designer¹¹ wants to be able to perform on the policy. In the first case (with an activation condition) it is possible to compute if the policy is active in a given situation and therefore plan the action for fulfilling or violating it. When computing all the activations may be too costly and the goal of reasoning is monitoring the fulfillment or violation of the policy, the second formalization, without activation condition, is the more efficient because it does not require to compute the activation of many policies.

In the ODRL 2.2 model it is possible to express *constraints* associated to the rules contained in one policy and *refinements* associated to the actions regulated by one rule (a duty, a prohibition, or a permission). Reading the documentation that provides the meaning of a prohibition or duty, the constraint can be used to express the activation condition, but again, like in OWL-POLAR, the temporal constraints between the satisfaction of the activation condition and the performance of the action regulated by the rule is not explicitly expressed in the model. Therefore, when the activation condition is a state of affairs the policy designer has to choose whether it is better to put the conditions in the constraint or in the refinement. Differently, when the activation condition of the norm is represented by a class of events, by using the ODRL 2.2 and OWL-POLAR models it is impossible to specify in the norm formalization the need to compute at run-time the value of the deadline and it is impossible to model those policies that when are activated generates *general* deontic relations.

¹¹The term “norm designer” refers to the person in charge of formalizing norms with a formal model.

3.5.3 Representing obligations and prohibitions

In this section, we describe how we can distinguish if a norm generates an obligation or prohibition and how to express them using the T-Norm model. In contrary with other approaches such as OWL-POLAR and ODRL, in the T-Norm model there is not a component or a predefined class that may be used to specify if the norm express an obligation or a prohibition. The advantage of this approach is that both obligation and prohibitions can be expressed starting from the same abstract norm and there is not need to formalize the semantics or a state machine (like in [60]) for obligations, another one for prohibitions, and others for other deontic concepts like permission, right, privilege, liability and so on. However, in this Section we focus only on obligation and prohibition.

In the T-Norm model the intuitive meaning of having an obligation or prohibition is that when something happens and certain conditions hold, an agent is obliged or prohibited to do something in a given interval of time. We can use few basic constructs and combine them in different ways to express the obligation to perform an action before a given deadline or the prohibition to perform an action within an interval of time. The main difference in representing a prohibition or an obligation is in the second THEN part of the norm. If the norm designer wants to formalize the obligation to perform an action, performing the regulated action must bring to the specification of the fulfillment of the deontic relation in the THEN part of the norm. The ELSE part have to be used to specify that in case on instance of the class of actions regulated by the policy cannot be performed before than a given event happens the deontic relation, representing the obligation, becomes violated. On the contrary, if the norm designer wants to formalize the prohibition to perform an action (described in the second ON part of the norm) in the specific interval of time, performing the action will bring to the violation of the deontic relation in the second THEN part. Once the prohibited action can no longer be performed (for example, the time interval has expired) the prohibition becomes fulfilled.

As we know from deontic logic literature[161] the expression “it is impermissible (IM) that p ” is defined as equivalent to “it is obligatory (OB) that not p ” ($IMp = def OB\neg p$). This implies that some norms may be either formalized as an obligation or as a prohibition.

When a norm is formalized with the T-Norm model and the activation of the norm brings to the creation of *general deontic relations*, it is very important to evaluate which of the two formalizations would be most cost-effective. That is because, as discussed below, every general deontic relation created by the activation of a norm, may in turn bring to the costly generation of many fulfillments and violations. For example, the norm “when the school bell rings, students should go back to the classrooms in five minutes” can be formalized as a norm that generates obligations or prohibitions. Suppose that the person in charge of formalizing the norm is only interested in computing the violations of the norm. In the first scenario, if we formalize the norm as a generator of obligations, when the activation condition is satisfied because the school bell rings, the norm generates a general deontic relation that will generate fulfillments for all those students who respect the school rule and go back to their classrooms, and violations for those students who did not fulfill the rule before the deadline. In the second scenario, it is possible to formalize such a norm as a generator of prohibitions by reframing it as follows “when five minutes have elapsed since the bell rang, students cannot remain in the courtyard”. The formalization of this norm is much easier and cost-effective as we only need to check the violations that are generated for those students who stay in the courtyard.

3.5.4 Temporal aspect of norms

The ability to represent time-constrained norms is one important characteristics of the T-Norm model. Unlike the OWL-POLAR model, in the T-Norm model can be used by the norm designer to easily represent any obligations containing deadlines (that are represented as time events) and prohibitions that holds for an interval of time. A norm governs a class of actions and, as can be seen from the abstract norm in Section 3.4, that class can be temporally constrained by specifying the BEFORE part and another class of events (?event3). The latter class can be specified using the TimeEvent class or the more generic Event class depicted in Figure 1.

The TimeEvent class is used for specifying a deadline for obligations or the instant at when a time interval for prohibitions ends. In this case the value of the deadline or the end of

3.5. Methodology

the time interval can be computed in the COMPUTE part of the norm as exemplified below. For example, in Norm01, an agent is obliged to perform the *paying* action before midnight (the deadline). In Norm02, the time interval in which Italian libraries cannot lend DVDs begins with the release of the DVD and ends after 2 years. Norm2 can be represented with the T-Norm model as follows:

Norm2

```
ON ?e1
  WHERE isReleased(?e1) AND object(?e1,?dvd) AND VideoObject(?dvd) AND
  place(?e1;Italy) AND atTime(?e1,?inst1) AND inXSDDateTimeStamp(?inst1,?t1)
THEN
  COMPUTE ?tend_n=?t1.year+2
  CREATE DeonticRelation(?dr);TimeEvent(?tev_end_n);Instant(?inst_end_n);
  ASSERT isGenerated(?dr,Norm2); activated(?dr,?e1); end(?dr,?tev_end_n);
      atTime(?tev_end_n,?inst_end_n);
      inXSDDateTimeStamp(?inst_end_n,?tend_n);
ON ?e2 BEFORE ?tev_end_n
  WHERE LendAction(?e2) AND object(?e2,?dvd) AND actor(?e2,?agent)
THEN violates(?agent;?dr); violated(?dr,?e2)
```

The CREATE and ASSERT parts of the norm above, which specify the characteristics of the time event used to constrain the class of actions governed by the norm, represent a prototype of what these two parts look like in all such type of norms.

On the other hand, if the temporal constraint (?event3) belongs to a generic Event class (or one of its subclasses), it is not necessary to compute anything. This means that the regulated action is temporally constrained by another generic class of events. For example, in the norm “You should pay the parking ticket before exiting”, there exists no deadline for the payment action, but the payment action must be performed before leaving the parking

area with one's car. This event should be specified in the WHERE part used to describe ?event3.

In literature there exist approaches, such as [127], in which they used temporal logics such as Linear Temporal Logic (LTL) for representing time-constrained norms. Nevertheless, using these approaches present some difficulties when it comes to the automatic reasoning on the evolution of the normative state from activated to fulfilled or violated.

3.5.5 Specific and General Deontic Relations

In the T-Norms model, a norm can create several deontic relations when the activation condition of the norm is satisfied. Such deontic relations may belong to one of the following two categories: *specific deontic relations* and *general deontic relations*.

A specific deontic relation is generated, when the regulated action should be performed by a specific agent, e.g. in Norm1, for each vehicle entering into the limited traffic area an obligation to pay for the owner of the vehicle is generated. In the specific deontic relations, the debtor, the owner of the vehicle, is known. Therefore, in case of any violation, the system can recognize who violated the deontic relation. The specific deontic relations generated by Norm1 have a debtor property that connects the deontic relation with the agent that is the owner of the vehicle: `ASSERT ... debtor(?dr, ?agent)...` This property has to be inserted in the ASSERT part of the norm for all those norms that generate specific deontic relations.

The second type of deontic relations is the general deontic relations. The main difference between general deontic relations and specific deontic relations is that in the first one we do not have any knowledge about the debtor of the class of actions regulated by the norm and the action can be performed by a set of agents, for example in Norm2 by all the people registered in one library. For that reason, we cannot have any predefined estimation about which agent is going to violate or fulfill the deontic relation. It is possible to have many violations and many fulfillments of the same deontic relation. For example Norm2 is activated for every new distribution of a DVD. The general deontic relation created by the

3.6. Conclusions

activation of Norm2 regulates the actions of lending such a DVD by all the Italian libraries, the debtor is not one specific agent.

Another significant difference between OWL-POLAR and the T-Norm model is in formalizing norms that generate general deontic relations. In OWL-POLAR a policy can only be activated for a specific agent therefore the type of norms that when activated regulate the actions of a set of agents cannot be represented. This is due to the design choice to propose a model for reasoning on policies that do not create deontic relations. In ODRL it has not been specified the mechanism for reasoning on policies activations.

3.6 Conclusions

In this paper, we introduced a methodology that explains how a norm designer can formalize norms written in a natural language into a machine-readable format by understanding the types of norms and choosing the proper model. As it is discussed in the previous section the norm can be: (i) a norm with or without an activation condition; (ii) if there is an activation condition it can be represented by a class of events or by a state of affairs; (iii) a norm can express obligations or prohibitions; (iv) a norm can regulate a class of actions that is time constrained or not; (v) finally a norm can generate specific or general deontic relations. In our future works we plan to extend the methodology by discussing the formalization of exceptions to norms and in particular of permissions and exemptions and the definition of institutional powers for manipulating norms.

4

How to Formalize Different Types of Norms in Multi-Agent Systems: a Methodology Focused on the T-Norm Model

4.1	Abstract	85
4.2	Introduction	85
4.3	Related works	90
4.4	Running examples	95
4.5	Types of Norms	95
4.5.1	Norms activated by an event or state of affairs or both	96
4.5.2	Norms defining specific or general obligation or prohibition	97
4.5.3	Norms expressing a temporal constraint for regulated actions	99
4.5.4	Norms that induce normative powers	100
4.6	The T-Norm model	102
4.7	Methodology	107
4.7.1	Using ontologies for modeling norms	107

4.7.2	Formalizing the activation condition	111
4.7.3	Representing obligations and prohibitions	114
4.7.4	Representing the temporal constraints of norms	116
4.7.5	Formalizing specific or general obligation or prohibition . .	119
4.7.6	Formalizing norms inducing normative power	120
4.7.7	Representing permissions and exemptions	122
4.8	Conclusions	125

Accepted Manuscript to be published by Springer Nature Journal: Roshankish S., Fornara N. (2024) How to Formalize Different Types of Norms in Multi-Agent Systems: a Methodology Focused on the T-Norm Model, accepted to be published in the peer-reviewed journal SN Computer Science, Springer Nature, 2024.

4.1 Abstract

In a world where many activities are carried out digitally, it is increasingly urgent to be able to formally represent the norms, policies, and contracts that regulate these activities in order to make them understandable and processable by machine. In multi-agent systems, the process to be followed by a person to choose a formal model of norms and transform a norm written in a natural language into a formal one by using the selected model is a demanding task. In this paper, we introduce a methodology to be followed by people to understand the fundamental elements that they should consider for this transformation. We will focus mainly on a methodology for formalizing norms using the T-Norm model, this is because it allows us to express a rich set of different types of norms. Nevertheless, the proposed methodology is general enough to also be used, in some of its steps, to formalize norms using other formal languages. In the definition of the methodology, we will explicitly state which types of norms can be expressed with a given model and which cannot. Since there is not yet a set of different types of norms that is sufficiently expressive and is recognized as valid by the Normative Multi-agent Systems (NorMAS) community, another goal of this paper is to propose and discuss a rich set of norms types that could be used to study the expressive power of different formal models of norms, to compare them, and to translate norms formalized with one language into norms written in another language.

4.2 Introduction

In a world where more and more activities are performed digitally by humans and software agents using the Internet, it is crucial to be able to *formally* represent the rules, norms, contracts and policies that govern the performance of these activities. These are activities of various types that have an impact on the digital life of citizens such as payments, access to online services and those related to creating, storing, using, exchanging and manipulating the enormous amount of digital resources that exist today, such as data spaces, images, videos, books, articles and so on.

4.2. Introduction

The formal representation of these rules is essential to make them readable and even better understandable by the machine and thus being able to provide useful services to those involved in the rules as parties and to those in charge of checking that they are respected. Indeed, it is important to keep in mind that these norms regulating the actions of autonomous agents cannot always be regimented (for example, it is very difficult to regiment obligations [56]), therefore it is possible for such rules to be violated and in certain applications it is crucial to be alerted whenever a violation occurs or a violation will occur as a result of the execution of a certain action.

Analysing the advantages of a formal specification of norms in more detail, we can see that when norms become machine-readable, it is possible to automatically analyze and query them as it is discussed in [120] where the PrivOnto ontology is used for analyzing 115 privacy policies. For example, it will be possible to search the set of resources on which it is possible to perform certain actions based on the customers' interests. When a policy is formalized with a model that has an operational semantic, it is also possible to provide services for compliance checking of policies [145, 125, 60, 62]. This functionality plays an important role, especially in domains in which the customers' sensitive data is collected because it helps to create a trustworthy environment for customers, this by providing a monitoring platform that they can use to see whether their privacy policies (norms) are violated or not. For instance, a customer can attach to one picture the prohibition to publish it on a public platform for advertisement and would like to monitor if the actions which are performed on the picture are compliant with this prohibition.

Another reason why it can be useful to specify norms with formal languages is that it may become easier for humans to understand their actual meaning, which is not always as immediate as it should be. For example, during the Covid-19 pandemic, it was not always easy to immediately understand what norms are in effect at any given time in a specific location and whether they entail obligations or prohibitions to perform actions. Another example of norms whose meaning and implications are not always immediately clear to the reader are the various privacy policies that regulate the processing of our data when we browse websites and use social networks. Users often accept such policies in order to use online services often without fully understanding what they mean, this is because they are

4.2. Introduction

too long or complex.

A considerable number of different models for the formal representation of norms and policies have been proposed in the academic literature in different fields of research such as the study of Normative Multiagent Systems, Digital Law, Privacy and Security, and Access Control for databases. For some of these models a logic-based semantics or an operational semantics has been proposed. In the approach described in this paper we will focus our attention on a subset of these models that are close to being able to be used in real applications on today's Web, especially those that make use of standard Semantic Web Technologies, a key characteristic for realizing interoperable systems and perform automatic reasoning with guarantees of decidability of the reasoning process.

One of these models, which owes its importance to being a W3C Recommendation, is the Open Digital Rights Language (ODRL 2.2). ODRL is a policy expression language that is specified through an Information Model¹ for specifying permissions, prohibitions, and obligations about the usage of digital assets and services. The semantics for the concepts and terms of the ODRL Information Model are specified in the core and in the common vocabularies (that are formalized in an OWL ontology)². Two other models for specifying norms and policies based on semantic web technologies are: the OWL-POLAR framework for semantic policy representation and reasoning [145] and the T-Norm framework for automatic monitoring of norms that regulate time constrained actions [62, 63]. As we will discuss in the rest of the paper, these are two complementary models of norms/policies whose operational semantics (the first computed using SPARQL-DL queries and the latter using production rules) can be used for different purposes: the former to do what-if reasoning (i.e. deducing what happens if an action is performed) and anticipating conflicts between policies; the latter to do monitoring and receive notifications if violations or fulfilments of norms occur.

The papers that introduced these models mainly consisted of presenting the components of the model and exemplifying its use by formalising a few examples of norms or policies. What is missing, however, is a *methodology* that explains what steps should be followed if

¹<https://www.w3.org/TR/odrl-model/>

²<https://www.w3.org/TR/odrl-vocab/>

4.2. Introduction

one wants to start from a norm written in a natural language (e.g., English) and be able to choose the model for its formalization and use it to arrive at the formal specification of the norm, which can then be used to reason about it and to evaluate its fulfillment or violation. In particular, in order to be able to decide which of the available models can be used for the formalisation of a given norm (e.g. a norm regulating access to restricted traffic zones in a city), it would be essential to know which types of norms can be formalised with a given model. But since there is no commonly accepted set of types of norms in the literature, papers presenting a given model do not always specify which types of norms can be expressed with that model and which cannot.

Contributions. Starting from these two shortcomings (the absence of a methodology and of a rich list of types of norms), in this paper we will give our contribution to the achievement of the following two goals.

The *first goal* of this paper is to propose a rich set of *norms types* that could be used to study the expressive power of different formal models of norms and to compare them. Knowing that a certain model is or is not capable of expressing a certain types of norms is fundamental for deciding which model is the best to adopt in a certain application context. For example, if the activation of a norm has to generate a specific obligation for a particular agent and with a precise deadline, it is necessary to choose a model of norms that allows us to express this temporal constraint and check its satisfaction. Secondly, once it is clear that a certain type of norm can be expressed in both model A and model B, it will also be possible to translate norms written in the first model into norms written in the second one. Thus making systems that use different norms models interoperable. This is a fundamental aspect in today's world where one software agent must be able to interact with multiple open socio-technical systems without having to be reprogrammed and where a system may need to monitor different set of norms that could be specified using different languages.

Our second objective is to propose a *methodology* that can be used by people to formalise norms that are written in a natural language with the chosen norm model. By using the term methodology, we mean a systematic method for dealing with a complex problem by specifying the various steps that need to be taken in order to achieve an end goal. This

4.2. Introduction

term is well suited to describing a complex task requiring interdisciplinary skills, in fact it requires a profound knowledge of the meaning of the policy one wants to start from, of the model one wants to use and of its semantics, furthermore since there is no one single formalisation possible it requires at each step to decide which of the alternative approaches to follow. The proposed methodology consists in first understand to which types the norm belongs. This step is important because once we know the types of the norm and assuming that we know for a certain model what types of norms it supports, we will be able to determine which models can be used to formalise the norm we started from. The second and more complex step of the methodology consists in coming to a proper formalization of the norm using the chosen model. In this paper, we will focus mainly on formalizing norms using the T-Norm norm model, this is because it allows us to express a rich set of different types of norms. However, it is important to emphasize that the proposed methodology is general enough to also be used, in some of its steps, to formalize norms using other formal languages that have some similarities with the T-Norm model, such as at least OWL-POLAR and ODRL.

This paper is an extended version of paper [137]. In this paper we introduce the Section “Type of Norms”(4.5) to describe a rich list of possible types of norms that can be found in normative systems. Moreover, referring to this new list of types of norms, we have clarified the steps of the methodology and we have added two subsections to represent norms inducing normative power among agents in multi-agent systems (Section 4.7.6) and a section to explain the formalization of exceptions that are used to specify permissions and exemptions (Section 4.7.7).

Organization. Section 4.3 discusses the problem of formalizing existing norms using formal models and presents relevant and recent papers presenting models for norms and policies specification in which Semantic Web Technologies have been used. In Section 4.4 two running examples are introduced. In Section 4.5, we will propose and explain a list of different types of norms. In Section 4.6, the T-Norm model is briefly presented. Then in Section 4.7, the proposed methodology for formalizing norms belonging to different types of norms is discussed. Finally in Section 4.8 we draw some conclusions.

4.3 Related works

In the multi-agent systems community, over the past two decades, many models of norms and policies³ have been proposed to regulate the behavior of autonomous agents [12, 37]. The Semantic Web and Linked Data community is also another active research community in studying the formalization of policy languages for regulating the use of digital assets [149], or for representing legal rules [66].

Norms, formalized with these models, can usually express various types of normative concepts, the most common of which are the notion of obligation, prohibition, and permission. These models (as will be discussed in this article) are generally capable of expressing only some of the different types of norms proposed in Section 4.5. For example, it may be the case that a model can be used to express obligations with a fixed deadline, but is not able to express an obligation whose deadline must be calculated when the norm expressing the obligation is activated.

Since a list of different types of norms that are commonly accepted by the research community does not yet exist, it is impossible for a paper proposing a model to clearly indicate the types of norms that can be expressed by that model. It is also rare to find papers or tutorials explaining a methodology that must be followed to formalize existing norms using a given formal model. That is a methodology that describes the steps to be followed by a person to choose the model of norms they want to use and guides them through the process of formalizing a norm written in a natural language using the selected model.

A paper that presents a framework and a methodology, termed LOGIKEY, was introduced recently in [22]. LOGIKEY stands for **L**ogic and **K**nowledge **E**ngineering Framework and **M**ethodology and it was introduced with the objective to support the practical development of computational tools for normative reasoning based on different formal methods. The LOGIKEY methodology consists of the following steps: (i) selecting a logic (like the dyadic deontic logic); (ii) selecting an ethico-legal domain of interest (like the German road traffic act or the GDPR); (iii) deploying the theory selected in the previous step in

³In some models the term policy is used, in others the term norm is used. For this reason, the two terms will be used as synonyms in this document.

4.3. Related works

practical applications to regulate the behavior of an intelligent autonomous system. This methodology is complementary to the one proposed in this paper. Indeed, the first step of the LOGIKEY methodology, which consists of choosing the logic to be used, bears some similarities with the methodology proposed here in which the choice of the model of norms that can be used to formalise a given norm is discussed. But in the LOGIKEY methodology they use classical higher-order logic (HOL) as their formal framework whereas we propose to use models of norms whose semantics is only partially based on description logic. Moreover, our methodology does not focus on choosing a formalism that is then used for a set of norms but explains how to formalise a set of norms of interest on the basis of their type.

Since we cannot compare the proposed methodology with other similar ones in the literature, we consider it useful to introduce in this section some models of norms that have some similarities with the T-Norm model of norms, which will be mainly used in this paper in the exemplification of the methodology. Thanks to these similarities, the proposed methodology can also be used, in some of its parts, for the formalization of norms using such models. In particular, we will focus in this section mainly on those models that use standard Semantic Web Technologies and/or rule-based systems for the formalization of certain components of the norms model or for reasoning about them.

The first norm models that used semantic technologies were the KAoS framework [156], the REI [93] policy language, and the Provisional Trust Negotiation (PROTUNE) framework [29]. These approaches are summarized and compared in [30], where the requirements for a policy framework are discussed and the various approaches are categorized, considering whether the policies are public or not. For example, for public policies, it is possible to use the KAoS and REI frameworks as only a one-step evaluation is needed to determine if the two policies are compatible. On the other hand, if a policy contains sensitive data, it is required to have *stateful* and *stateless* negotiation protocols for further security concerns.

A well-known policy language based on semantic web technologies, which is a W3C Recommendation since 15 February 2018, is the Open Digital Rights Language (ODRL

4.3. Related works

2.2)⁴. It is a policy expression language that can be used to represent permitted, prohibited, and obliged actions over a certain asset. ODRL policies may be limited by constraints (e.g., temporal or spatial constraints). ODRL was originally (in 2001) an XML language for expressing digital rights, that is, digital content usage terms and conditions. ODRL 2.2 is a Policy Language formalized in RDF with an abstract information model specified by an ontology. It has no formal semantics, so compliance checking of policies written with this language cannot be performed automatically. An interesting attempt to give a formal semantic to ODRL 2.1 policies is presented in [149]. Some extensions of ODRL have been proposed to overcome some of its limits. In particular, in [60] an extension of the ODRL Information Model has been proposed together with a set of state machines used for describing the evolution in time of the deontic state of obligations, prohibitions, and permissions. Another extension of ODRL is presented in [44] to model both regulatory policies (in the form of nested permissions, prohibitions, obligations, and dispensations), and business policies via discrete permissions. A policy written with that extension of the ODRL language is then translated into an Institutional Action Language (InstAL)[124] policy and thanks to its formal semantics, expressed in Answer Set Programming, it is possible to automatically check compliance and also provide an explanation of the aspects of the policy that brings to the non-compliance. In [72] a specific use case drawn from the social networks field is used to validate the expressiveness of the ODRL 2.0 model.

Other two interesting proposals of a policy/norm model and framework, which are based on semantic web technologies, are OWL-POLAR [145] and T-Norm [62, 63]. Those policies/norms models and their expressivity will be discussed in Section 4.6. An interesting aspect that differentiates the two models is the way in which these models define mechanisms to reason about policies to test whether agents' behavior satisfies them or not. In the OWL-POLAR a query answering mechanism (DL-safe) has been used to check if any action that happened satisfies the policies. In the T-Norm model, a rule-based approach is used that brings the generation of several deontic relations used to represent obligations and prohibitions generated by the activation of norms. In addition, the T-Norm model makes it possible to formalize the temporal constraints that exist between the activation of

⁴<https://www.w3.org/TR/odrl-model/>

4.3. Related works

a norm and the class of actions regulated by the norm.

Other interesting models of norms that, like the T-Norm model, are rule-based are: the one proposed by Garcia-Camino et al. [68] where rules are operationalized using the JESS a rule engine for the Java platform; and the one proposed in [11] where reasoning on norms is realized with DROOLS a business rule management system. Another interesting proposal of a language for the specification of legal text is the OASIS standard LegalRuleML⁵ [15], which defines a rule interchange language for the legal domain that is formalized using RuleML.

In the models that we used in this paper to represent our methodology (such as the T-Norm model or OWL-POLAR), we assume that norms are already selected and we only need to formalize them. However, there exist norm selection approaches that assume the availability of a collection of norms, emphasizing the consideration of moral values in the selection process. In [146] the authors advance the Normative multi-agent systems literature by formally defining the problem and proposing its encoding as a linear program, making it controllable to automated solutions. The research introduces three norm relationships—generalization, exclusivity, and substitutability and considers norm representation power, cost, and associated moral values as explicit preference criteria. The paper emphasizes the importance of including moral values in decision-making, which is a novel contribution in the context of normative systems. In practical terms, the study’s findings have implications for dynamic organizations or evolving societies that already have norms in force. The decision-making process involves considering existing norms alongside a new set of candidate norms. The proposed optimization-based framework provides a systematic approach to determine the optimal norm system, balancing conflicting objectives and constraints. Overall, the paper contributes to enhancing transparency and decision processes in practical tools, paving the way for future research on norm decision scenarios and the interplay between norms and moral values.

Two years later, in [147] the authors propose a qualitative approach to norm selection, focusing on inferring a norm system ranking based on value preferences. This approach

⁵<https://www.oasis-open.org/committees/legalruleml/>

4.3. Related works

avoids the quantitative approaches of previous methods and introduces a novel method for transforming qualitative preferences into norm system preferences. They make two major contributions: a novel ranking system allowing qualitative assessment of norm systems based on promoted values and an encoding method to solve the norm selection problem as a linear program. Some other studies address the challenge of coordinating multi-agent systems through the synthesis of norms. In [109], a novel mechanism named IRON (Intelligent Robust On-line Norm synthesis mechanism) is introduced and designed for the on-line synthesis of norms. IRON aims to produce conflict-free norms for effective coordination without lapsing into over-regulation, ensuring that the synthesized norms are necessary and concise. The norm synthesis problem, which involves determining a set of norms to avoid conflicting states, has been approached through both off-line and on-line strategies. The off-line approaches focus on synthesizing norms at design time, requiring detailed knowledge of the system. In contrast, on-line approaches, such as norm emergence, aim to regulate a system at runtime. However, these approaches face drawbacks like sensitivity to initial conditions and assumptions about agent collaboration. Against this backdrop, IRON is proposed as an on-line synthesis mechanism that not only ensures the effectiveness and necessity of norms but also addresses over-regulation by generalizing norms. Another framework focused on off-line synthesis is introduced in [110]. This paper explains a framework for the off-line synthesis of Evolutionarily Stable Normative Systems (ESNS) within multi-agent systems. The focus is on coordinating agents in multiple interdependent situations that cannot be easily identified in advance. The framework is based on evolutionary game theory and employs simulations and domain information to automatically enumerate potential conflict situations. Norms are then synthesized through an evolutionary process, leading to sets of codependent norms that effectively coordinate agents in various situations. The paper empirically evaluates the framework in a simulated traffic domain, demonstrating its ability to synthesize ESNSs that successfully avoid conflicts in numerous interdependent traffic situations.

4.4 Running examples

In the remainder of this paper, we will use various examples of norms to exemplify the types introduced and the procedure leading to their formalization with a given formal model. Two concrete examples will be particularly relevant and are as follows.

The first example (which we call Norm1) is inspired by the law regarding access to limited traffic area in Milan city, its formalization in natural language is: “*when an agent enters in the limited-traffic area of Milan, between 7:30 and 19:30, they have to pay 5 euro before 24:00 on the day of entry*”⁶.

The second example (which we call Norm2) is the rule that must be followed by libraries in Italy regarding the lending of DVDs, its natural language formalization is: “*Italian libraries cannot lend DVDs until 2 years are passed from the distribution of the DVD*”⁷.

We are going to use these examples in the following sections as well together with some other original examples that have been used in the related works.

4.5 Types of Norms

As mentioned in the introduction, one of the two objectives of this paper is to introduce an extensive list of the various types of norms that can be found in existing regulatory systems governing human relations from laws, to contracts, to policies for the use of services or resources. Our focus will be on *regulative norms*, i.e. norms that regulate the performance of actions by making them obligatory, prohibited or permitted. In this section we will introduce and explain what characterises these different types of norms.

Being able to define a list of types of norms that is accepted by the scientific community is important because it could lead to the possibility to study which types of norms can be formalised with a certain model and which cannot. Secondly, it could lead to the possibility of recognising that a certain existing rule (e.g. an article of a law) belongs to a specific type

⁶<https://www.comune.milano.it/aree-tematiche/mobilita/area-c>

⁷According to Art. 69 c.1 of the Copyright Law (22-4-1941, no. 633)

4.5. Types of Norms

and it is important in order to then be able to decide which model to use for its formalisation among all those with which that type of norm can be formalised. For example, if we are able to recognize that a norm is activated by an event and when it is activated generates a specific obligation to perform a certain action before a deadline, it is necessary to choose a model capable of expressing all these characteristics in order to express the norm into a machine-readable format.

Based on our research and the study of related works [12, 37, 46, 145, 91, 121, 142], we have identified the following four main types for norms:

1. Norms activated by an event or a state of affairs or both;
2. Norms defining general or specific obligations or prohibitions;
3. Norms expressing a temporal constraint for the actions that regulate;
4. Norms that induce normative powers.

It is important to mention that a norm defined in a certain normative system (like for example the European General Data Protection Regulation (GDPR)) may belong to more than one of the types of norms defined here. It means for example that it is possible that a norm that is activated by an event will create specific obligations characterized by a temporal constraint for the regulated action that should happen before a specific deadline.

4.5.1 Norms activated by an event or state of affairs or both

The first important distinction between different types of norms can be made on the basis of the type of condition that must be satisfied in order for a norm to become active.

On one hand, there are norms that are activated by the occurrence of an event belonging to a specific class of events (which is described in the norm). When such an event happens, the activation of the norm creates a specific type of deontic relation that obliges certain agents to do something before a specific point of time (deadline) or prohibits them from doing it for a certain interval of time. For example, Norm1 belongs to this type, in fact when drivers enter the limited traffic area of Milan (activation event), they become obliged to do

4.5. Types of Norms

a payment of a certain amount (regulated action) before the specified deadline (midnight on the day of entry). Norm2 also belongs to this type, in fact, from the moment a DVD begins to be distributed on the market (activation event), libraries are prohibited from lending it (regulated action) for two years (interval of time when the prohibition is active). In both examples, there exists a class of actions (where the class of actions is a subclass of the class of events), such as entering a limited traffic area or the distribution of a DVD on the market, such that when an action belonging to that class is performed, it leads to the activation of the respective norm.

On the other hand, there are norms that are activated when a certain state of affairs is satisfied and deactivated when that state of affairs is no longer satisfied. For instance, if we take into account the following norm discussed in [142] that regulates littering in a public park: *“agents should not drop litter in the park as long as a rubbish bin is 50 meters away from them”*. This norm is activated for an agent when there is a rubbish bin close enough to it and it is deactivated (or we can say it expires) when the agent moves away from a rubbish bin. In this example, the distance between an agent and the rubbish bin is the condition (or the state of affairs) for becoming obliged to use it or for activating the prohibition to drop litter in the park.

There are also norms that are activated upon the occurrence of an event only if certain states of affairs hold. For example, the obligation to pay for access to a city’s restricted traffic zone only if you enter on weekdays.

In Section 4.7, we will describe how to model a norm, whether activated by the occurrence of an event or the satisfaction of a state of affairs. We will see how this distinction has implications for the choice of the best model to formalize such norms.

4.5.2 Norms defining specific or general obligation or prohibition

When a norm is activated (whether by the occurrence of an event/action or by the satisfaction of a state of affairs), it creates an obligation or a prohibition to perform an action that

4.5. Types of Norms

belongs to a specific class of actions. For example, the obligation to pay a certain amount of money or the prohibition to lend a specific DVD.

The obligation or prohibition created by the activation of a norm can apply either to a *specific* agent, who is called the *debtor* of the specific obligation or of the specific prohibition or to a more *general* set of agents, e.g. all the agents playing a given *role* at the time of the activation (we speak about general obligations or prohibitions).

Norm1 belongs to the first category of norms: when a driver enters a limited traffic zone, the owner of the car (a specific agent known at the time of activation) becomes obliged to make the payment before the deadline, and if the payment is not made on time, the owner of the car is the agent who violates the obligation.

Norm2 belongs to the second category of norms. When a DVD is distributed, the norm is activated and it starts to hold the prohibition for all the Italian libraries to lend the specific DVD until the interval of 2 years elapses. The debtor of the prohibited action (lending DVD) is not a specific agent, it is a set of agents, i.e. the agents who play the librarian role in Italian libraries. When this type of norm is activated, there is no specific agent playing the role of the debtor of the obligation or prohibition. The specific agent becomes known in case of violation of the prohibition because he or she is the actor who performs the prohibited action, i.e. the librarian who lend the DVD before the two years passed.

Therefore we can observe that a norm that creates an obligation (like Norm1) or a prohibition that applies to a specific agent can bring the generation of one fulfillment or one violation. Differently, a norm that creates an obligation or a prohibition (like Norm2) that applies to a group of agents can bring the generation of many violations or fulfillments, e.g. in the DVD example, more than one librarian can violate the prohibition.

The recognition of the difference between these two types of norms is important because there are certain models that can be used to formalize a norm if and only if the debtor of the action regulated by the norm is a specific agent that becomes known at the time of the activation of the norm. Such a model cannot be used for the formalization of those norms whose specific debtor is unknown at the time of their activation.

4.5.3 Norms expressing a temporal constraint for regulated actions

An important and widespread type of norms involves specifying whether a particular action must be carried out (obligations) or prohibited (prohibitions) within a certain time interval. In particular, when the norm expresses an obligation, the obliged action must be performed after the activation of the norm and before a specific instant of time that is called the deadline. In this type of norm, when the deadline is elapsed the obligation can no longer be fulfilled and becomes violated. This type of norm is very common in normative systems because the presence of the deadline allows for an expectation of the behavior of the agents subject to these norms.

When the norm expresses a prohibition, the performance of the regulated action is prohibited from its activation until a certain instant of time, after which the prohibition can no longer be violated and becomes fulfilled. For this type of norm, the deadline for obligations or the end of the time interval in which a prohibition is in force may be calculated based on the time at which the norm is activated. Recognizing that a norm specifies a deadline (when it expresses an obligation) or an interval of time (when it expresses a prohibition) is important because it determines the model of norms that should be used for its formalization.

Norm1 and Norm2 belong to this type of norm. Norm1 expresses that when a driver enters a restricted traffic zone, the time instant of this action is very important for calculating the deadline of the payment obligation. Having an obligation to pay without a deadline would not make sense. Norm2 expresses that the release date of a DVD is the starting point of the two-year prohibition on lending DVDs. It is also possible to have another type of prohibition: the prohibition to perform an action without an associated time interval, in this case the prohibition to perform the action will be active forever.

When rules express an obligation to perform an action within a deadline, it is possible to identify two sub-types of such norms based on whether or not the obligation persists after the deadline has passed as it is discussed in [46]. In a certain type of norm, the deadline

4.5. Types of Norms

for the obligation is such that if an agent does not perform the required actions before the deadline, the obligation is violated and the agent can no longer fulfill the obligation. For instance, if a student submits assignments after the deadline, they will not be graded by the teacher. On the other hand, there are norms in which the obligation persists even if the deadline is not met. In the example of the restricted traffic zone, if a driver does not make the payment by the deadline, it does not mean that they are no longer obliged to pay. Instead, they will receive an additional payment obligation on top of the first one, which is the fine for not meeting the deadline. This latter type of regulations are known as *standing obligations*. In some scenarios, it is also possible that the same obligation with a deadline is repeated over a period of time. We call these deadline obligations as *repetitive obligations*[46]. An example of such obligations are the monthly payments to be made by mobile phone subscribers.

In Section 4.7 we will describe how to formalize norms having temporal constraints in detail.

4.5.4 Norms that induce normative powers

Another type of norm is the one where the norms are able to represent the *normative power* of agents or with a different name used in literature the *institutionalized power* of agents [91].

In [121] the notion of *normative power* is defined as follows: in a normative system, an agent has a normative power if the agent has the ability to create, modify or delete certain norms, and those changes to the norms are recognized by the other agents. This notion of power has been introduced to represent the fact that an agent can modify only a subset of a normative system. This notion of normative power that focuses on modifying only norms runs the risk of not emphasizing that the notion of normative power is related in general to the creation or modification of the normative state of affairs involving agents, i.e. when one changes their rights or obligations not only their norms [91].

4.5. Types of Norms

Given that, in many norms models, by activating a norm an agent may be able to create obligations or prohibitions for other agents, this means that certain types of norms are able to give a particular power to the agents playing a certain *role*, i.e. the role that an agent has to play for being able to perform the action that activates the norm. For example, in a certain university, the professors supervising a PhD student (they have the specific role of supervisor) have the normative power to request the student to write a paper for a conference, but they do not have the power to request the student to do the shopping for them. This normative power could be represented by a norm that is activated with the request to write a paper. When such a norm is activated then an obligation for the student is created. In [45] these kinds of norms are called *norms in their role definition*. In this paper when we speak about the particular type of norms that induce normative powers, we are considering this extended notion of normative power.

When one wants to understand whether a particular norm is a norm that induces normative powers, it is important to check whether it has the ability to give normative power to certain agents who play specific roles. One approach to figuring out whether norms induce agents' normative power or not is to look for the notion of "power" or similar notions in the description of the norms. For example in the norm, "*The University has the power to oblige students to participate in lectures*", the word "power" already exists in the formulation of the norm. Therefore, by introducing that norm in a normative system the agents are aware that the university acquires such a normative power to create this obligation for its students.

When the notion of *power* is not explicitly present in the formulation of the norm, distinguishing between the norms that give the agents a normative power and those that do not induce a normative power may become difficult. This is because it is possible that the notion of power is hidden in the description of norms. It means that, despite the absence of the notion of *power* in the description of the norm, if in a norm *Agent A* (usually thanks to their roles) may take an action that creates a deontic relation (an obligation or a prohibition) for another *Agent B* in a multi-agent system and *Agent B* together with all the other agents in the system recognized that they should fulfill such a deontic relation that is created by *Agent A*, we say that *Agent A* has a normative power over *Agent B*. This

happens for example in the following norm: “*When professors ask their Ph.D. students to write a paper for a conference, the students are obliged to do it before the deadline of the conference*” where there is not any notion related to power in its description. However, by reading the norm we understand that there is an agent (Agent A) with the role of professor that (by performing the action of asking) may create an obligation for another agent (Agent B) having the role of student. Therefore this norm represents the professors’ normative power who can oblige a student to write a paper.

We will discuss in detail how it is possible to model this kind of norm in different circumstances in Section 4.7.6.

4.6 The T-Norm model

The T-Norm model can be used to formalize a precise and rich set of *types of norms* that regulate the interactions between autonomous agents. Namely (as we will further discuss in the paper) the model can be used to formalize those norms with an activation condition expressible as a class of events; the norms that generate general or specific obligations or prohibitions to perform (or not to perform) actions that can be constrained to happen before something else happens; exceptions to those norms and to obligations and prohibitions (i.e. exemptions and permissions respectively); and norms that induce a normative power. Once a set of norms is formalized using the T-Norm model, a specific *framework* can be used to automatically check if the agents’ behavior conforms or does not conform to the given set of norms. This is done by monitoring the evolution of the state of those sets of norms as time passes, events occur and autonomous agents perform actions. The framework for norms monitoring has been proposed by taking into account the *operational semantics* of the T-Norm model. The model, its operational semantics, and the framework were introduced in [62, 63].

The T-Norm model captures the following intuitive meaning of norms: whenever a particular *activation condition* is satisfied (i.e. an event that belongs to a particular class of events occurs) a *deontic relation* (general or specific) is created for regulating (oblige or

4.6. The T-Norm model

prohibit) the performance of a class of actions by certain agents. In turn, every time an action belonging to the *class of the regulated actions* is executed before a certain event happens (for example a certain temporal event representing a deadline) and the deontic relation represents an obligation it will be *fulfilled*, while it will be *violated* if it represents a prohibition. On the contrary, when an action belonging to the class of regulated actions can no longer be performed (for example because the deadline has expired) and the deontic relation represents an obligation it will be *violated*, while if the deontic relation represents a prohibition it will be *fulfilled*.

In order to formally describe such a dynamic behavior, the abstract model of a norm cannot only consist of a set of facts (like it is in many models of norms and policies, e.g. ODRL, OWL-POLAR [145], and the model proposed in [11]). In all these models the intrinsically dynamic nature of norms is described in their semantics or is left to their intuitive meaning described in the text. The T-Norm model allows us to specify how the performance of certain actions or the occurrence of certain events will change the state of the interaction among agents. Therefore the basic building blocks of the T-Norm *Abstract Norm* are *rules* of the form ON . . . THEN . . . ELSE⁸. The *Abstract Norm* has not a pre-defined deontic type, as will be discussed in Section 4.7, it is those who formally specify a norm who will decide whether the norm activation creates obligations or prohibitions. In the T-Norm model, a generic *Abstract Norm* has the following form:

```
-----  
Abstract Norm  
-----  
  
1:  NORM Norm_n  
2:  [ON ?event1 WHERE conditions on ?event1  
3:    THEN  
4:      COMPUTE]  
5:      CREATE DeonticRelation(?dr);  
6:      ASSERT isGenerated(?dr, Norm_n); [activated(?dr, ?event1);]
```

⁸The ON clause has been chosen instead of the more common IF clause to highlight that the part after THEN is executed when a particular event or action occurs and not simply when a condition is satisfied

4.6. The T-Norm model

```
7:      ON ?event2 [BEFORE ?event3 WHERE conditions on ?event3]
8:          WHERE actor(?event2,?agent) AND conditions on ?event2
9:      THEN ASSERT fulfills(?agent,?dr); fulfilled(?dr,?event2) |
10:          violates(?agent,?dr); violated(?dr,?event2)
11:      [ELSE ASSERT violates(?agent,?dr); violated(?dr,?event3) |
12:          fulfills(?agent,?dr); fulfilled(?dr,?event3]
```

In the proposed model the first (optional) ON...THEN component (line 2,3) is used for expressing those norms that have an activation condition. The second ON...THEN component (line 7,9) is used for expressing that when a specific action, which belongs to the class of actions regulated by the norm, is performed (before something else occurs) there will be a fulfillment or a violation. In alternative, the ELSE part of the second rule (line 11) will be followed when an action that belongs to the class of the regulated actions can no longer be performed.

The *formulas* used in the *Abstract Norm* are conjunctions (in the WHERE part) or sequences (in the CREATE and ASSERT part) of *atomic assertions* written using the classes (unary predicates starting with a capital letter) and the properties (binary predicates starting with a lowercase letter) defined in the *T-Norm Ontology* depicted in Figure 1⁹. Variables (starting with '?') refers to individuals. Variables used in the WHERE parts of the norm for expressing conditions on events can be used freely and have to be bound to individuals in the *State Knowledge Base* (where the interaction among agents is represented) for the conditions to be met. In the WHERE parts, it is also possible to compare the value of a variable with a constant value using any of the symbols {<, >, =, ≠, ≤, ≥}. A constant is a numerical value or an individual in the ontology. Variables that appear in the ASSERT part of a norm must have been introduced previously in one of its ON or CREATE parts. In the COMPUTE part some values can be calculated (for example the deadlines) using the value of previously introduced variables¹⁰.

⁹The T-Norm ontology in OWL is available at <https://raw.githubusercontent.com/fornaran/T-Norm-Model/main/tnorm.owl>

¹⁰The choice of using conjunctions or sequences of atomic assertions (analogously to what is proposed in OWL-POLAR to express the various components of their norm model) has the advantage of avoiding

4.6. The T-Norm model

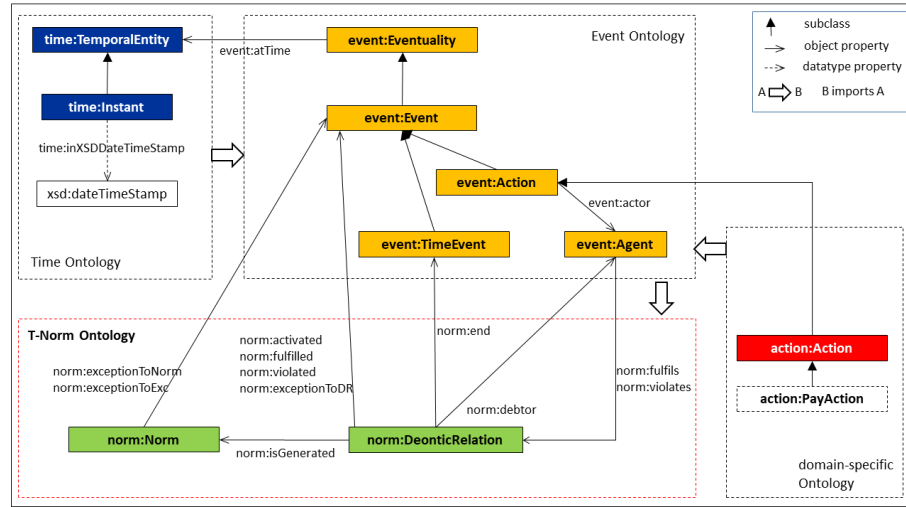


Fig. 4.1: The T-NORM Ontology and its connections with other ontologies.

By using the T-Norm model it is possible to formalize exceptions, that can be used for modeling both *permission*, as an exception to prohibitions, and *exemption* as an exception to obligations. By using the model it is possible to specify different types of exceptions. The first is an exception to the activation of norms. When certain peculiar conditions are satisfied on the event activating the norm, the resulting deontic relation has not to be generated. For example “firefighters do not have to pay the train ticket”. The second type of exceptions are those to deontic relations, i.e. when specific conditions on the regulated event are fulfilled the generation of violation/fulfilment is blocked. For example, in a library, the lending of a certain book is prohibited to everyone except teachers. There is also a third type of exceptions, these are those always related to deontic relations (like the second type) but which are triggered by an event that is not the one regulated by the norm. This third type of exception can be formalized only for those norms that generate *specific* deontic relations.

An exception is expressed in one of the following ways on the basis of its type:

requiring the user of the model to learn a specific formal language, once written those expressions can be easily and automatically translated into the conditions or actions of production rules or into SPARQL queries)

4.6. The T-Norm model

Exception Type1

```
-----  
EXCEPTION TO Norm_n  
  ON ?event1  
  WHERE conditions on event1  
  THEN exceptionToNorm(Norm_n,?event1)
```

Exception Type2

```
-----  
EXCEPTION TO Norm_n  
  ON ?event2  
  WHERE conditions on event2 AND isGenerated(?dr, Norm_n)  
  THEN exceptionToDR(?dr,?event2)
```

Exception Type3

```
-----  
EXCEPTION TO Norm_n  
  ON ?event_n  
  WHERE conditions on event_n AND isGenerated(?dr, Norm_n) AND  
    NOT fulfills(?agent,?dr) AND NOT violates(?agent,?dr)  
  THEN exceptionToDR(?dr,?event_n)
```

In order for exceptions to inhibit norms, they must be evaluated before the norms. In the T-Norms model where norms and exceptions are translated into production rules to be computed, the rules of exceptions have higher priority than those of norms. In Section 4.7.7 we will explain how to formalize these types of exceptions in detail by using some examples.

4.7 Methodology

In this section, we describe the various steps of the procedure to be followed to transform a norm written in a natural language (for example in English) into a norm written using a formal machine-readable language like the T-Norm model. As will be discussed, some steps of the described procedure can also be used to formalize norms using the OWL-POLAR model or the ODRL policy expression language. Starting from a norm expressed in natural language, following each step of the methodology, the *Abstract Norm*, introduced in Section 4.6, is made at every step more concrete to the point of being the formalization of the norm from which the process started.

4.7.1 Using ontologies for modeling norms

As discussed in the Introduction, we decided to propose a methodology that can be used to formalize norms by using models based on Semantic Web Technologies. In particular, in those models the activation conditions of the norms, the class of actions that is regulated by the norm, the expiration conditions, and the temporal constraints to the performance of the actions regulated by the norm are specified using the classes (aka concepts) and the properties defined in ontologies encoded using one of the semantic web languages, namely the W3C Web Ontology Language (OWL 2) or RDF-Schema. An important advantage of using those well-known standard languages is that they can easily be used by those who need to formalise norms.

In the **first step** of the procedure, it is required to: *identify the events, the actions, the activation and expiration conditions, and the temporal constraints described in the norm and decide which OWL or RDF-Schema ontologies can be used to represent them*. Such ontologies can be defined from scratch or better, in a re-usability perspective, through the re-use of existing ontologies that can then be adapted to express the concepts and properties present in the norm under consideration.

When the ODRL policy language is used for the formalization of one policy, the terms

4.7. Methodology

in the policy can be expressed using the *ODRL Common Vocabulary*¹¹ that provides a list of actions and properties for representing statements about the usage of content and services (for example it defines actions like Display, Distribute, Derive and properties like payAmount and spatialCoordinates).

When the norm is formalized using the T-Norm model, three *classes of events or actions* should be specified using ontologies:

- The class of *events* that represent the *activation condition* of the norm. The variable ?event1 in line 2 of the *Abstract Norm* refers to an individual that belongs to the activation condition class and such a class is described in the WHERE part using the classes and properties defined in the adopted ontologies;
- The class of *actions regulated by the norm*. The variable ?event2 in lines 7,8 of the *Abstract Norm* refers to an individual that belongs to the class of actions regulated by the norm. Such a class is described in the WHERE part (line 8) using the classes and properties defined in the adopted ontologies;
- The class of *events* defined for *constraining the performance of the actions regulated by the norm*. One action, belonging to the class of the regulated ones, should or should not occur before an event belonging to this constraining class. The variable ?event3 in line 7 of the *Abstract Norm* refers to an individual that belongs to the constraining class. As in the previous two points, in the WHERE part (line 7) the constraining class is specified using the classes and properties defined in the adopted ontologies.

The ontologies selected for the formalization (each one referred to as a *domain-specific Ontology* in Figure 3.4) should be imported into the *Event Ontology* which is formalized using the OWL 2 language. Therefore even if in principle it is possible to use several different ontologies for specifying the class of actions and their properties inside one T-Norm norm, for compatibility reasons, we suggest using ontologies that are compatible with OWL ontologies.

¹¹<https://www.w3.org/TR/odrl-vocab/#vocab-common>

4.7. Methodology

When a norm is formalized with OWL-POLAR policy model, the class of actions regulated by the policy is specified in a similar manner as with the T-Norm model. Due to substantial differences between the two models, in OWL-POLAR it is not required to describe a class of events as activation condition but it is possible to specify a condition on the state of the world (such as “when there is a fire risk” or “if a room contains patients”). Furthermore, it is not required to specify a temporal constraining class of events but it is necessary to specify an expiration condition (such as “there is no longer a fire risk” or “the room is empty”).

We will now exemplify the formalization of the two classes of events necessary for the formalization of Norm1. Norm1 is activated every time a vehicle enters the restricted traffic zone of the city of Milan. We assume that the `RestrictedTrafficAreaAccess` is an OWL class of actions, `vehicle` and `owner` are two properties; the first has a domain `RestrictedTrafficAreaAccess` class, the second has a domain of class `Vehicle`. Those classes and properties are defined in an OWL 2 domain-specific Ontology. The class of events that activates Norm1 can be specified in the following way ¹²:

Norm1: Activation Condition

```
NORM  Norm1
      ON ?e1 WHERE RestrictedTrafficAreaAccess(?e1) AND vehicle(?e1,?v)
      AND owner(?v,?agent) AND atTime(?e1,?inst1)
      AND inXSDDateTimeStamp(?inst1,?t1) AND ?t1.time(>07:30:00
      AND ?t1.time(<19:30:00
```

In the previous expression the variable `?agent` is introduced because it will be used in the second part of the norm to recognize who fulfills or violates the norm. The properties used to express conditions on when the event occurs are defined in the Time Ontology in OWL ¹³ which is imported in the *Event Ontology* as it is represented in Figure 3.4.

¹²where `?event1` is shortened to `?e1`

¹³<https://www.w3.org/TR/owl-time/>

4.7. Methodology

The class of actions regulated by Norm1 is the payment of 5 euro before 24:00 on the day of entry. For formalizing it we can for example use the `PayAction` class defined in the *Schema.org*¹⁴ vocabulary, which has an OWL version. Schema.org represents an interesting attempt to realize a lightweight ontology that can be reused in different applications and in particular it is used for the specification of structured data on the Internet and on web pages and is used by main search engines.

As mentioned earlier, the class of events described with the variable `?event3` has the role of constraining the time interval in which the action belonging to the class of actions regulated by the norm shall or shall not be performed. In Norm1, the time interval when the payment action should be performed is constrained by a deadline (below referred with the variable `?paymentDeadline`), i.e. the payment action must occur before 24:00 on the day of entry into the limited traffic area. The formalization of norms where `?event3` is a time event are discussed in Section 4.7.4. However, in the formalization of other norms the time constraint could be expressed with any class of actions (e.g. the payment must be made before leaving the restricted area) and in this case we will need to use another ontology to represent that class. The class of actions regulated by Norm1 can be specified in the following way (where `?e1` and `?agent` are the variables introduced in the specification of the activation condition of Norm1 in the previous ON clause):

Norm1: Regulated Pay Action

```
ON ?e2 BEFORE ?paymentDeadline
  WHERE PayAction(?e2) AND reason(?e2,?e1) AND recipient(?e2,MilanCity)
  AND price(?e2,5) AND priceCurrency(?e2,euro)
```

¹⁴[https://https://schema.org/](https://schema.org/)

4.7.2 Formalizing the activation condition

The goal of this section is to explain the **second step** of the methodology. It consists in recognizing whether a norm has an *activation condition* or not and if it does, how to formalize it by taking into account its type, i.e. whether the norm is activated by the occurrence of an event that belongs to a *class of events or actions* or by the occurrence of a certain *state of affairs*.

The question to ask in order to recognise whether a rule has a trigger condition is: under what conditions is there an obligation or prohibition for someone to perform a certain action? If these conditions correspond to the occurrence of an event or action then the norm can be modelled with the T-Norm model and it is important to underline that in the T-Norm model, the activation condition of a norm cannot simply be a description of a state of affairs. If these conditions relate to the fulfilment of a certain state of affairs then the norm can be modelled with the OWL-POLAR or ODRL model.

A very common objection to the differentiation between events and states of affairs such as the one proposed here is that an event leads to a new state of affairs and that therefore the new state of affairs can be used as a condition instead of the event. For example the effect of entering in the limited-traffic area of Milan is the state of affairs of being inside the area. But this event-state substitution is not always possible because for certain norms the activating event has crucial characteristics (such as an instant of time when it happens) that cannot be replaced by a state of affairs.

The reason why, in norm models, these two different types of activation conditions are treated differently is mainly due to their ontological difference: an event happens at a specific point in time and when it has happened it can no longer be retracted. Differently, there are states of affairs that can be satisfied for a certain interval of time and become unsatisfied in another interval (as for example the condition “*it is a weekday*”). This is a crucial difference, indeed in the T-Norm model, any satisfaction of the activation condition leads to the permanent creation of deontic relations. This permanent creation is required when the newly created deontic relation regulates a class of actions that should or should not be performed in an interval of time and when the deontic relation itself can

4.7. Methodology

generate many violations and fulfillments, as it is discussed below when Norm2 will be formalized. On the other hand, in the OWL-POLAR and ODRL models, the satisfaction of the activation condition simply leads to an activated policy for a specific agent, and when the activation condition is no longer satisfied or the expiration condition becomes satisfied, the policy becomes inactive.

By using the T-Norm model, in order to recognize the activation condition in the text of a norm, we have to look for the events or actions that when happen induce the model to create certain obligations or prohibitions. When an event that belongs to the class of events described in the activation conditions actually occurs a new *deontic relation* expressing the obligation or prohibition to do a certain class of actions is created and, if applicable, the values of the time constraints relating to the execution of that action are calculated. The temporal relation between an event or action that satisfy the activation condition and the action that should or should not be performed is crucial: the activation condition must be satisfied before the obligation or the prohibition to perform a certain class of actions starts to hold. The instant of time at which the activation condition of a norm is satisfied by an event or action is very important because it can be used to calculate the deadline of obligations generated by the norm or the instant of time at which a prohibition ceases to subsist. For example in Norm1, the activation condition is represented by the class of actions regarding entering the Milan limited traffic zone and it is used for computing the deadline for the payment, its formalization is available in the previous Section. In Norm2, the activation condition is given by the class of actions with which a DVD distribution is initiated and can be formalized, using the classes and properties of OWL ontologies, in the following way:

Norm2

```
1: ON ?e1
2:  WHERE isReleased(?e1) AND object(?e1,?dvd) AND VideoObject(?dvd)
3:  AND place(?e1;Italy) AND atTime(?e1,?inst1)
```

4.7. Methodology

4: AND inXSDDateTimeStamp(?inst1,?t1)

When the activation condition of a policy describes a state of affairs, the OWL-POLAR model can be used for their formalization. Examples of this type of policies are: “*a person is obliged to leave a location when there is a fire risk*” or “*when a person has a child which is under 18 they have to pay their tuition*”[145], where the activation condition is the part after the word “*when*”. In the OWL-POLAR model, a policy is activated for a specific agent when the world state is such that the activation condition holds for that agent and the expiration condition does not hold. Therefore, at the time of activation, it is necessary to know the specific agent for whom the policy is being activated. As we will further discuss in Section 4.7.5, this is not the case when the activation of a policy leads to the creation of *general deontic relations*. Another interesting aspect of the OWL-POLAR policy model, is that in this model there is not an explicit representation of the requirement that the activation condition must be satisfied before a policy could become fulfilled or violated. This requirement is only expressed in the description of the process for reasoning about policies activation.

In the ODRL 2.2 model, the Rule class is the parent of the Permission, Prohibition, and Duty classes and it represents their common characteristics. It is possible to express constraints associated with the Rules contained in one policy and refinements associated with three parts of the rule: the actions regulated by one Rule, its Asset Collection, and its Party Collection. In the ODRL Information Model 2.2 there is a textual specification of the meaning of a prohibition and a Duty, and one can see the satisfaction of constraints can be considered as an activation condition of the rule. Therefore, when the activation condition is a state of affairs, the policy designer can choose whether it is better to put the conditions in the constraint or in the refinement of an ODRL policy. But, as in the case of OWL-POLAR, the time constraints between the satisfaction of the activation condition and the execution of the policy-regulated action are not explicitly expressible with the ODRL model and it is impossible to specify in the policy formalization the how to calculate at run-time the value of deadlines. It is also impossible

to formalize those policies that when they are activated generate *general* deontic relations that will be discussed in Section 4.7.5.

By analysing the numerous examples of policies formalised with the ODRL 2.2 model, we made an interesting observation: when the activation condition in the text of a rule describes a state of affairs, one should ask whether this condition should actually be formalised as an activation condition or whether it is better to formalise it as a set of conditions delimiting the class of actions regulated by the policy. For example, consider the following conditional norm “*it is prohibited to litter as long as there is a rubbish bin within x meters from an agent*” (which is discussed in [142]). The condition of being within x meters of a rubbish bin may be modeled as an activation condition but it can also be considered as a condition that constrains the class of actions regulated by the policy, i.e. littering when the actor of the action is within x meters from the rubbish bin. When this last approach is adopted, the policy can also be modeled with the T-Norm model because there is not anymore a state of affairs as an activation condition. The choice between using the first or second formalization depends on the type of reasoning that the norm designer¹⁵ wants to be able to perform on that policy. In the first case (when the state of affairs is formalized as an activation condition) it is possible to compute if the policy is active in a given situation and therefore plan the action for fulfilling or violating it. When computing the satisfaction of all the activation conditions of the policies may be too costly or the goal of reasoning about policies is monitoring their fulfillment or violation, the second formalization (without activation condition) is the more efficient because it does not require computing the activation of the policies.

4.7.3 Representing obligations and prohibitions

In this section, we explain the **third step** of the methodology, by clarifying: (i) how to recognize whether a norm generates an obligation or a prohibition; (ii) how to express obligations and prohibitions using one formal model.

¹⁵The term “norm designer” refers to the person in charge of formalizing norms with a formal model.

4.7. Methodology

By reading the text of a norm it is quite intuitive to recognise its deontic component, usually the verbs “*have to*” or “*must*” or “*is obligatory*” are used to express obligations. Conversely, the negation of these verbs or the verb “*cannot*” are used to express prohibitions.

Sometimes, however, it may be convenient to formalise a norm containing an obligation as a prohibition and vice versa. In fact, as we know from deontic logic literature[161] the expression “it is impermissible (IM) that p ” is defined as equivalent to “it is obligatory (OB) that not p ” ($IMp = def OB\neg p$). In particular, when the activation of a norm brings to the creation of *general deontic relations* (see Section 4.7.5 for more details), it is very important to evaluate which of the two formalizations would be most cost-effective. That is because, as discussed below, every general deontic relation created by the activation of a norm, may in turn bring to the costly generation of many fulfillments and violations. For example, the norm “when the school bell rings, students should go back to the classrooms in five minutes” can be formalized either as a norm that generates obligations or as a norm that generates prohibitions. Suppose that the person in charge of formalizing the norm is only interested in computing the violations of the norm. In the first scenario, if we formalize the norm as a generator of obligations when the activation condition is satisfied because the school bell rings, the norm generates a general deontic relation that will generate fulfillments for all those students who respect the school rule and go back to their classrooms, and violations for those students who did not fulfill the rule before the deadline. In the second scenario, it is possible to formalize such a norm as a generator of prohibitions by reframing it as follows “when five minutes have elapsed since the bell rang, students cannot remain in the courtyard”. The formalization of this norm is much easier and cost-effective as we only need to check the violations that are generated for those students who stay in the courtyard.

Once it has been decided whether to formalise a particular norm as an obligation or as a prohibition, if it has been chosen to use the OWL-POLAR or the ODRL model, it is enough to explicitly indicate if the policy is (or contains) an obligation or a prohibition. Differently from those two models, in the T-Norm model there is not a component or a predefined class that may be used to specify whether the norm expresses an obligation or

a prohibition. The advantage of this approach is that both obligation and prohibitions can be expressed starting from the same abstract norm and there is no need to formalize the semantics or the state machine for obligations, another one for prohibitions, and others for other deontic concepts like permission, right, privilege, liability and so on, as it is proposed in [60].

In the T-Norm model the intuitive meaning of having an obligation or prohibition is that when something happens and certain conditions hold, an agent is obliged or prohibited to do something in a given interval of time. We can use a few basic constructs and combine them in different ways to express the obligation to perform an action before a given deadline or the prohibition to perform an action within an interval of time. The main difference in formalising a prohibition or an obligation lies in the choice of the `fulfills` or `violates` assertions to be included in the last **THEN ... ELSE** part of the abstract norm presented in Section 4.6.

If the norm designer wants to formalize the obligation to perform an action, performing the regulated action must bring to the specification of the fulfillment of the deontic relation by a specific agent in the **THEN** part of the norm. The **ELSE** part has to be used to specify that in case an action belonging to the class of actions regulated by the policy cannot be performed before a certain event happens, the deontic relation (representing the obligation) becomes violated. On the contrary, if the norm designer wants to formalize the prohibition to perform an action in a specific interval of time, performing the action will bring to the violation of the deontic relation in the **THEN** part. Once the prohibited action can no longer be performed before a certain event happens (for example, the time interval has expired) the deontic relation (representing the prohibition) becomes fulfilled.

4.7.4 Representing the temporal constraints of norms

The **fourth step** of the methodology consists in recognising whether the norm regulates an action that must, or must not, be performed before:

1. an event/action occurs (i.e. “*pay before leaving a car park*”), or

4.7. Methodology

2. a given instant of time has elapsed (i.e. “*pay before the end of year 2023*”), or
3. an instant of time, whose value is calculated at the time of activation of the norms, has elapsed (i.e. as in Norm1 “*pay before midnight on the day of entry*”).

This can be done by searching the text of the norm for the words “*before*” or “*until*” or the indication of precise time intervals where the regulated action should or should not be performed.

In order to formally express such a constraint the T-Norm model has to be used. In fact, the ability to represent temporally constrained actions in norms is one of the distinguishing features of the T-Norm model that differentiates it substantially from the OWL-POLAR and ODRL models. Indeed in these two models, it is impossible to express deadlines for obligations, especially those whose value is computed at run-time.

A completely different time constraint that can be expressed in OWL-POLAR is the *expiration condition* of a policy that, when it is satisfied, makes the obligation or prohibition inactive (for example, the obligation to leave a place ends when the risk of fire ceases). Similarly, when ODRL is used, it is possible to formalize a constraint for the performance of an obliged or prohibited action by using the `dateTime leftOperand`, but its semantics is not well documented and examples of its use express only permission and no obligation or prohibition.

Once it is realised that the T-Norm model must be used to formalise the norm under consideration, it is necessary to proceed by specifying the part of the *Abstract Norm* (see Section 4.6) that follows the BEFORE clause. Here it is required to specify the description of the class of events that constrains the performance of the actions regulated by the norm by specifying to which class the variable `?event3` belongs and what the values of some of its properties must be. Based on the type of constraint specified in the norm, it must be specified that the variable `?event3` belongs to the `TimeEvent` class or the more generic `Event` class (see Figure 3.4).

The `TimeEvent` class is used for specifying that `?event3` is a time event that happens at a specific instant of time, which represents a deadline for an obligation or the instant of time when a prohibition ends. For example, in Norm1, an agent is obliged to perform the

4.7. Methodology

paying action before midnight (the deadline) of the day on which the norm was activated. In Norm2, the time interval in which Italian libraries cannot lend DVDs begins with the release of the DVD (activation of the norm) and ends after 2 years. Following all the steps of the methodology explained so far, Norm2 is a prohibition activated by an event that can be represented with the T-Norm model as follows¹⁶:

Norm2

```
1: ON ?e1
2:   WHERE isReleased(?e1) AND object(?e1,?dvd) AND VideoObject(?dvd)
3:   AND place(?e1;Italy) AND atTime(?e1,?inst1)
4:   AND inXSDDateTimeStamp(?inst1,?t1)
5: THEN
6:   COMPUTE ?tend_n=?t1.year+2;
7:   CREATE DeonticRelation(?dr);TimeEvent(?tev_end_n);Instant(?inst_end_n);
8:   ASSERT isGenerated(?dr,Norm2); activated(?dr,?e1);
9:   end(?dr,?tev_end_n); atTime(?tev_end_n,?inst_end_n);
10:  inXSDDateTimeStamp(?inst_end_n,?tend_n);
11: ON ?e2 BEFORE ?tev_end_n
12:   WHERE LendAction(?e2) AND object(?e2,?dvd) AND actor(?e2,?agent)
13: THEN violates(?agent;?dr); violated(?dr,?e2)
```

The COMPUTE, CREATE and ASSERT components are important parts of the norm above, which are used to specify the characteristics of the time event used to constrain the class of actions governed by the norm. Their formulation in the above norm represents a prototype of what these components look like in all kinds of norms of this type. In particular, the time event, represented with the variable ?tev_end_n (instead of ?event3), is associated to a specific instant of time by using the atTime property whose value is computed when the norm is activated.

¹⁶The formalization of Norm1 using the T-Norm model is available in [62]

4.7. Methodology

The `Event` class (or one of its sub-classes) is used for specifying that the temporal constraint of the norm belongs to a generic class of events. This means that the regulated action is temporally constrained by another generic class of events. For example, in the norm “*you have to pay the parking ticket before leaving*”, there exists no deadline for the payment action, but the payment action must be performed before leaving the parking area with one’s car. Therefore, in the ‘conditions on ?event3’ part of the *Abstract Norm* the event of leaving the parking should be specified.

In literature there exist other models, such as [127], in which time-constrained norms are represented using temporal logics, such as the Linear Temporal Logic (LTL). However, using these approaches present some difficulties when it comes to automatically reasoning on the evolution of the normative state from activated to fulfilled or violated.

4.7.5 Formalizing specific or general obligation or prohibition

The **fifth step** of the methodology consists in recognizing whether the norm under consideration is a norm that defines *specific* or *general* obligation or prohibition, that were introduced in Section 4.5.2. Namely, it is important to distinguish between two types of norms: those that can be activated for a *specific* set of agents, this means that at the time of the activation the specific debtor of the activated obligation or prohibition is known; and those that can be activated for a *general* set of agents, whose specific name is unknown at the time of the activation.

The fact that the norm to be formalized belongs to one or the other type has an impact on the choice of the normative model to be adopted. In particular, if the norm belongs to the first type (such as Norm1), both the T-Norm and the OWL-POLAR model can be used; if the norm belongs to the second type (such as Norm2) only the T-Norm model can be used for its formalization. Regarding the ODRL model, it can only be used when the debtor (that is ODRL is called the assignee) of the obligation or prohibition is already known before the policy is activated.

We will now better explain how to proceed in formalising the norm. If the norm defines a *specific* obligation or prohibition and the OWL-POLAR model is chosen, it is important to include the $?x$ variable, used in OWL-POLAR to represent the policy addressee, in the policy activation condition as well. This is required because the policy can only be activated for all those *specific* agents for which the activation condition is satisfied. Thus, when the condition is satisfied by the state of the world, there will be one or more substitutions of the $?x$ variable present in the activation condition with specific agent names.

Differently, if the T-Norm model is adopted, we have that the norm can create many *specific deontic relations* with a known debtor, one, each time its activation condition becomes satisfied by the execution of an action or the occurrence of an event. For example in Norm1, for each vehicle entering into the limited traffic area an obligation to pay for the owner of the vehicle is generated. This is expressed in the formalization of Norm1 by using, in the ASSERT part, the debtor property to connect the generated deontic relation with the owner of the vehicle that entered in the limited traffic area (i.e. by using the following formula: `debtor(?dr, ?agent)`). The debtor property has to be specified in the ASSERT part of the formalization of all those norms that generate specific deontic relations.

If the norm belongs to the second type, only the T-Norm model can be used for its formalization. For example, when Norm2 is activated, i.e. when the DVD is distributed, the activation condition is not satisfied for specific agents. In this case the agents who will be able to violate the prohibition to lend the DVD are the actors of the lending action, i.e. the agents working for Italian libraries. This type of norm cannot be formalized using the OWL-POLAR model because the satisfaction of the activation conditions does not provide substitutions for the $?x$ variable that is used for representing the policy addressee. This limitation of the model is mainly due to the design choice to propose a model for reasoning about policies that do not create deontic relations when policies are activated.

4.7.6 Formalizing norms inducing normative power

The **sixth step** of the methodology consists in understanding whether the norm implies normative power or not, this by taking into account what is explained in Section 4.5.4.

4.7. Methodology

Norms represent a normative power when, thanks to the norm, one agent can trigger the creation of an obligation for another agent. For example, in the following norm (that we call Norm3) “*when professors ask their PhD students to write a paper for a conference, the students are obliged to do it before the deadline of the conference*”, the agents playing the role of professors have a specific normative power over their PhD students. Conversely, if the activation of the norm creates an obligation for the same agent whose action led to the activation of the norm, the norm does not give any normative power to agents with a given role. For example, Norm1, according to which if an agent enters a restricted traffic zone, they become obliged to pay 5 euro to the City of Milan, does not represent any normative power.

After understanding whether or not the norm represents a normative power, the norm designer must figure out how this power is represented and, based on this, select the appropriate model to represent the norm. As described in Section 4.5.4, there are two types of such norms. Firstly, there are norms that have the notion of *power* explicitly in the description of the norm, and secondly, there are norms that induce normative power for agents who have a certain role but do not explicitly mention the notion of normative power in their text.

If the notion of power is explicit in the text of the norm, a model of the notion of normative power, such as that proposed in [121], is required for the formalization of the norm. For example, this would happen for the formalization of the following norm: “*When Professor John asks a PhD student to write a conference paper and they have the normative power to oblige his PhD students to write a conference paper, the student becomes obliged to write the paper*”.

On the other hand, if the norm is stated in such a way that it does not require an explicit formalization of the concept of normative power as in the case of Norm3, then it is possible to model it with a norms model, such as the T-Norm, without using a model of normative power. Indeed, Norm3 can be formalized using the T-Norm model as:

Norm3

```

-----
ON ?e1
  WHERE WritingPaperRequest(?e1) AND Professor(?p) AND Student(?s)
  AND actor(?e1,?p) and receiver(?e1,?s) AND object(?e1,?paper)
  AND Conference(?conf) AND target(?paper,?conf)
  AND hasDeadline(?conf, ?deadline)
THEN
  COMPUTE ?t_end=?deadline
  CREATE DeonticRelation(?dr);TimeEvent(?tev_end_n);Instant(?inst_end_n);
  ASSERT isGenerated(?dr, Norm3); activated(?dr, ?e1);
    debtor(?dr, ?s); end(?dr, ?tev_end_n);
    atTime(?tev_end_n, ?inst_end_n);
    inXSDDateTimeStamp(?inst_end_n, ?t_end);
ON ?e2 BEFORE ?t_ev_end
  WHERE WritePaperAction(?e2) AND actor(?e2, ?s) AND
    object(?e2, ?paper) AND target(?paper, ?conf)
  THEN ASSERT fulfills(?s, ?dr); fullfilled(?dr, ?e2)
  ELSE ASSERT violates(?s, ?dr); violated(?dr, ?t_ev_end)

```

4.7.7 Representing permissions and exemptions

In this section, we describe how to formalize those norms that express the deontic notion of *permission* and *exemption*. The meaning of having a permission to do something has been extensively researched in the literature, and several forms of permission have been analyzed. In [26, 25], they consider permissions as an exception to prohibitions, which means that the agent's behavior does not count as a violation and thus there will be no penalty if the permitted action happens. In many normative systems there is a default assumption related to permissions and prohibitions, i.e. when it is necessary to explicitly specify permission to perform an action, it is because a prohibition to perform that action already exists or it is assumed to exist. Indeed, it is possible to distinguish among two

4.7. Methodology

types of permissions: *weak permission* and *strong permission*. The important distinction between strong and weak permission has been addressed in [73]. If there is no prohibition to do an action, it means that we have weak permission to do that action. On the other hand, if there exists explicit permission for doing a prohibited action, we call it strong permission. In a similar situation for obligations, if an agent has strong permission to derogate an obligation, it is called an *exemption*.

The notion of permission is formalized in different ways in the various models of norms considered in this work. In the T-Norm model the notion of *exception* can be used for formalizing both strong permissions (exceptions to prohibitions) and exemptions (exceptions to obligations) as will be presented below. Differently, in OWL-POLAR weak permissions are simply represented with a *P* indicating that the policy is a permission, all other policy parameters are those already discussed for the specification of obligations or prohibitions. The model does not provide a way to indicate that a permission allows an action that is prohibited. Similarly, in ODRL, permissions are formalised within a policy using the permission property and “a Permission allows an action, with all refinements satisfied, to be exercised on an Asset if all constraints are satisfied and if all duties are fulfilled”¹⁷. For instance, it is possible to specify that an agent is allowed to play a game for a certain time interval and it is impossible to indicate what is the precise prohibition to which an exception is being created.

When a prohibition or obligation is formalised using the T-Norm model, it is possible to introduce exceptions to it using one of the following types, whose abstract model was introduced in Section 4.6:

1. Exception to norms activation;
2. Exception to deontic relations (that were generated by the activation of a norm) due to some particular conditions on the event that is regulated by the norm;
3. Exception to specific deontic relations due to an event that is not the one regulated by the norm.

¹⁷<https://www.w3.org/TR/odrl-model/#permission>

4.7. Methodology

Exceptions of the **first type** are used to block the activation of a norm in certain specific cases concerning the conditions of activation of the norm. For example, one may wish to inhibit the activation of Norm1 if the vehicle entering the limited traffic area is an ambulance. In the specification of the exception, it is enough to write the further conditions that ?e1 must satisfy, knowing that when the exception is evaluated, the other conditions on ?e1 contained in Norm1 will also be considered. The exception is able to inhibit the activation of the norm because the constructs specifying exceptions are all evaluated before the constructs used to represent the norms in a system. The exception to Norm1 described above is formalised as follows:

```
EXCEPTION TO Norm1
ON    ?e1 WHERE Ambulance(?v)
THEN exceptionToNorm(Norm1,?e1)
```

Where the specific exception to Norm1 for the specific event is created if the vehicle that satisfies the activation condition of Norm1 (represented with the variable ?v), by entering in the limited traffic area, belongs to the class Ambulance.

The **second type** of exception is used to inhibit the generation of the violation or fulfilment of deontic relations even if the action regulated by the norm is performed or not performed. For example, an exception to the prohibition expressed by Norm2 may be to give teachers permission to borrow DVDs from the library for educational purposes even if the two years have not elapsed. In the specification of the exception, it is enough to add that the recipient of the lending action introduced in the formalisation of Norm2 (see Section 4.7.4) by using the variable ?e2, belongs to the class Teacher. This exception is expressed with the T-Norm model as follows:

```
EXCEPTION TO Norm2
ON    ?e2
WHERE recipient(?e2,?agent2) AND Teacher(?agent2) AND isGenerated(?dr,Norm2)
THEN exceptionToDR(?dr,?e2)
```

4.8. Conclusions

The **third type** of exception is used to inhibit the fulfillment and violation of *specific deontic relations*. Due to the fact that these exceptions are brought about by events that are not the one regulated by the norm, they differ from those of the second kind. For instance, an exception to the following Covid-19 norm: “a person who has a positive swab to Covid-19 cannot leave the house for the next 15 days”, is: “*if the house is on fire then everybody is allowed to leave it*”. This exception is activated by an event (the house is on fire) that is different from the action that is regulated by the norm (leaving the house). We model this exception in the following way:

```
EXCEPTION TO NormCovid
ON ?en
WHERE Fire(?en) AND place(?en,?house) AND residence(?house,?agent) AND
      isGenerated(?dr, NormCovid) AND activated(?dr,?e1) AND
      affectedPerson(?e1,?agent) AND NOT fulfills(?agent,?dr) AND
      NOT violates(?agent,?dr)
THEN exceptionToDR(?dr,?en)
```

4.8 Conclusions

Nowadays, the importance of formalizing the rules, norms and policies that govern activities of human and software agents in order to make them understandable and processable by machines, encourages many research communities to propose formal models or languages to reach this goal. In this paper, we proposed a methodology that explains how a norm designer can formalize norms written in a natural language into a machine-readable format by understanding the type of the norm under consideration, choosing the appropriate model, and using it correctly.

The presentation of this paper has two objectives. Firstly, proposing and discussing a methodology that can be used by people to translate norms written in natural language into a language specifically designed for the formal specification of norms. The second goal is to propose and discuss a rich set of norm types that could be used in the steps

4.8. Conclusions

of the proposed methodology, to study the expressive power of different formal models of norms, to compare them, and to design methods for translating norms written in one formal language into norms written in another formal language enhancing the interoperability of normative systems.

The proposed methodology is relevant mainly for two reasons. The first is that it introduces some criteria for the choice of the model of norms to be adopted for the formalization of a given norm. For example, one of the criteria discussed is whether the norm is activated by the occurrence of an event or by the occurrence of a state of affairs; another criterion is the presence in the norm of a deadline within which a certain obliged action must be performed. The second is that it is not at all obvious how a norms model can be used to formalize existing norms written in natural language and how to choose between different possible formalizations of the same norm, e.g. deciding whether it is better to formalize a norm as a prohibition to do an action or as an obligation to do the opposite action. The value of the proposed methodology has been highlighted by discussing the formalization with the T-Norm model of two different types of running examples, by presenting the formalization of a norm that induces a normative power and by exemplifying how permissions can be formalized as exceptions to prohibitions.

An interesting characteristic of the proposed methodology is that it is difficult to automate because it requires a deep understanding of the meaning of a norm and requires the user to be able to choose between different ways of formulating the same norm e.g. as an obligation or as a prohibition as discussed in Section 4.7.3.

A limitation of the proposed methodology is that the evaluation of its comprehensibility and feasibility of application for the formalisation of different types of norms has been carried out by a restricted group of researchers. The difficulty in this type of evaluation lies in the fact that in order to be able to use a formal model of norms based on semantic web technologies, it is necessary to have good knowledge in different fields, from the legal one (to understand the meaning of the norm) to the computer field and in particular in the field of normative systems and semantic web technologies (to be able to use formal ontologies), which is not easy to find in a significantly large group of people.

4.8. Conclusions

Another limitation of the proposed methodology is that it focuses only on manually transforming norms in natural language into machine-readable norms. We have not investigated how this process can be, at least in part, automated by, for example, trying to extract the meaning of certain parts of the text (e.g. the activation condition of a norm) with machine learning techniques, especially those based on Natural Language Processing techniques as it is discussed by the “AI and Law” community [160].

In our future work we plan to extend the methodology by discussing the formalization of norms and investigating the possibility of adding more types to our type selection. For example, we plan to investigate the formalization of sanctions for norms violations and the formalization of constitutive norms. In addition, we plan to test our methodology with a significant number of developers (students or colleagues who are familiar with this field of research) to check the feasibility of our approach and after receiving their feedback, apply the required modification for the improvement of the methodology. In our future work we plan also to study the applicability of the proposed methodology to other models of norms and in particular to those proposed in the legal field where the “AI and Law” community has proposed interesting legal ontologies and legal rule languages. We also plan to investigate how to extend the T-Norm model so that it can be used for formalizing norms activated by states of affairs.

Declaration

Funding: This work is funded by the Swiss National Science Foundation (SNSF) grant no. 200021_175759/1.

Conflicts of interest: The authors have no conflicts of interest to declare that are relevant to the content of this article.

5

Exploration of Norms and Policies in Digital Fashion Domain Using Semantic Web Technologies

5.1	Abstract	131
5.2	Introduction	131
5.3	Methodology	133
5.4	Semantic Web Technologies in Digital Fashion	134
5.4.1	Semantic Web and Semantic Web Technologies	135
5.4.2	Related Works Using Semantic Web Technologies in Digital Fashion	137
5.5	The Open Digital Rights Language	138
5.5.1	The ODRL Information Model	138
5.5.2	Examples of ODRL policies for the fashion domain	139
5.6	Monitoring Norms and Policies	141
5.7	Conclusion	143

Published Manuscript: Roshankish, S., Fornara, N. (2022). Roshankish, S., Fornara, N. (2021). Exploration of Norms and Policies in Digital Fashion Domain Using Se-

mantic Web Technologies. In: Soares, M.M., Rosenzweig, E., Marcus, A. (eds) Design, User Experience, and Usability: Design for Contemporary Technological Environments. HCII 2021. Lecture Notes in Computer Science, vol 12781. Springer, Cham. https://doi.org/10.1007/978-3-030-78227-6_28

5.1 Abstract

Nowadays, one of the problems with policy notices of fashion companies is that they are not provided in a machine-readable format; therefore, they cannot be searched and monitored by computers. The fact that most fashion brands transform their sales into digital format magnifies the importance of automation in policy management for the future of digital business in general and of fashion companies in particular. In this paper, we explore the use of privacy policies of companies in the digital fashion domain not only to protect customers' data but for make it feasible for companies to understand their customers' needs easier and faster. We describe how the Open Digital Rights Language (ODRL), a W3C recommendation for expressing policies using Semantic Web Technologies, can be applied in the field of digital fashion. We then discuss the required components for making it possible to use such a policy language for monitoring and enforcement services.

5.2 Introduction

Today, the number of companies using digital technologies to communicate with their customers is increasing expeditiously. After the COVID-19 pandemic, traffic to the top 100 fashion brands' owned websites increased up to 45 percent in Europe in April 2020 [7]. Meanwhile, Human-Computer Interaction (HCI) studies are trying to enhance this connection in e-commerce by developing new technologies. Talking about e-commerce, the global fashion industry with a revenue of 606 billion dollars by 2020 has a major role in the economy of every country [96]. However, in digital fashion, there are numerous privacy concerns for companies and brands in the digital transformation of their industry [138, 120].

Digital fashion communication embraces communication with brands designers and clothes online [69] while using official media and technological channels such as websites, social media, Augmented Reality applications [144], Haptic Technologies [122], and many others to reach customers [95]. In this paper, we tackle customer privacy as one of the major challenges in digital fashion. Although all companies are obliged to inform their clients

5.2. Introduction

about their policies before using their data, still we face a big gap in this area. For example, if somebody wants to shop on the Zalando website, they should create an account and accept the terms and conditions provided by Zalando before buying their favorite shoes. It means that they should read over 19,000 words (48 pages in Zalando UK) and consent to practices described in it[3]. That may be the reason why we are not surprised by the result of the survey done by Deloitte in USA showing that 91 percent of users consent to their legal rights without reading them [6]. If we calculate the time to read this policy as 250 words per minute [105], which is a common reading rate for people with a high school education [36], it takes over 75 minutes to read the policy notes of Zalando. The results show that if Internet users want to read their online privacy policies word by word each time they visit a new website, it costs billions of dollars nationally for the country[105].

In the last decades, discussion about privacy policies has been one of the major subjects in e-commerce applications. In the digital fashion, this issue could be crucial since companies could collect rich and accurate personal information about customers. For example, the fashion label Tommy Hilfiger introduced new technology in 2018 using smart chips in its products to collect information about how often the items would be worn by the customers [2]. Though Tommy Hilfiger ensured that customers' data is encrypted and cannot be accessed without any permission, it still concerns privacy issues that must be addressed.

Since 2018, all companies must comply with the requirements of the General Data Protection Regulation (GDPR) regarding the collection and handling of customer data. For instance, brands should inform customers about how data will be collected, stored, and used, or whether it will be shared with any third parties. Even having customers permission to use their data, they have the right to access their collected data or even delete them [1].

Although GDPR requires the costumers' consent for fashion brands to collect their data (same for the Tommy Hilfiger case), as mentioned before, most people do not read the terms and conditions of the websites before registering their information. Therefore, there is no control over whether or not companies are violating customer's privacy policies and we believe that transforming this process, thanks to the use of machine-readable formats,

helps both parties understand their rights in addition to improving customers' satisfaction.

If we want to solve this problem, we need to address the various issues that cause it. First of all, companies should start expressing their privacy policies in a machine-readable format. It is undoubtedly difficult to take such a measure, and in this paper, we propose to do that by using an existing policy expression language and Semantic Web Technologies. Having done so, we need to create a software infrastructure that can handle the meaning of policies when it is specified in a format that computers can process. To achieve this, the notation used to express privacy policies must have a formal semantics that allows computers to infer conclusions from the data collected on the behaviour of the involved parties and from the policies adopted. Therefore, we will discuss the required components for making it possible to use such a policy language for monitoring and enforcement services and we will briefly present two works that attempt to solve such a problem.

This paper is organized as follows. In Section 5.3 the methodology adopted in this paper is described. In Section 5.4, the Semantic Web Technologies used in this paper are introduced. In Section 5.5 the ODRL policy expression language, which is W3C Recommendation since 2018, is introduced. In Section 5.6, we address the importance of policy monitoring and outline the necessary components that a system should have to accomplish such monitoring automatically, including some references to work proposing solutions to this problem.

5.3 Methodology

In this paper, we propose an approach for modeling and monitoring the policies of brands and companies in the digital fashion domain not only to protect customers' data but to make it feasible for companies to understand their customers' needs easier and faster.

In Artificial Intelligence literature, there exist various languages that can be used for the specification of policies using Semantic Web Technologies [98]. One of them is the W3C Recommendation Open Digital Rights Language (ODRL 2.2). It is a policy expression language that can be used to represent permitted and prohibited actions over a certain

5.4. Semantic Web Technologies in Digital Fashion

asset, and obligations that should be met by various stakeholders. This language can be used to express the deontic aspect of fashion policies in a machine-readable format.

In addition, to be able to unambiguously formalize the actions that should or should not be performed over fashion products, for example, e-commerce actions related to clothes realized with specific materials, it is required to formalize those actions and the properties for describing fashion products using Semantic Web Technologies. In particular, the definition of shareable fashion ontologies and knowledge graphs is fundamental. In literature, there are some examples of ontologies used in the fashion domain[28, 150, 36] but it is not yet clear if they are expressive enough to be used for the specification of the actions regulated by policies and for taking advantage of automatic reasoning in the fashion domain.

Finally, in order to provide monitoring services for the evolution in time of deontic policies (e.g. computing the fulfillment or violation of obligations and prohibitions) a formal semantics for the ODRL policy language is required. One attempt to extend the ODRL language and to specify its formal semantics has been proposed in [60]. In this approach, different ontologies (such as a domain-specific ontology, the OWL Time Ontology, and an Event Ontology [54]) have been used. Therefore, in order to use this model for fashion-related policies, the domain-specific ontology must be a rich and expressive fashion ontology.

5.4 Semantic Web Technologies in Digital Fashion

Nowadays, norms and policies for regulating the use of personal data and digital assets in digital business in general and in the fashion industry, in particular, are only expressed in a human-readable format. This means that customers should read the policy terms of the companies before they are able to order their products online and understand all the implications of their actions. In this paper, we propose to use Semantic Web technologies to express those policies and automatically reason about their meaning. This section is a brief introduction to the Semantic Web technologies used in the next sections and explains

why the Semantic Web can have a significant impact on the fashion industry today and in the near future.

5.4.1 Semantic Web and Semantic Web Technologies

There are growing appeal for using Semantic Web in many research areas, since Tim Berners- Lee introduced the Semantic Web (or Web of Data) in 1999. The semantic Web is an extension of current World Wide Web (WWW), in which information is given a well-defined meaning that helps computers and people communicate and understand each other's needs [78, 23].

The important goal of the Semantic Web is to help advanced applications improve their search, navigation, and evaluation by making knowledge widely accessible. One of the key benefits of the Semantic Web is enabling computers to read the information in a structured format. The strength of the Semantic Web lies in modeling the knowledge (our privacy policies) in such a way that computers can draw conclusions from given information.

To have such flexibility we can use Semantic Web Technologies to translate the data to the formal computers' language. All the technologies presented in this paper, have been defined under the lead of the World Wide Web Consortium (W3C).

The eXtensible Markup Language (XML) and the Resource Description Framework (RDF) are the two important technologies for developing the Semantic Web. XML is a text-based Markup Language which can be used to label structured data by using tags [148]. The meaning of each tag is determined by the mutual agreement of those who are using a specific XML language [10], but XML documents do not have formal semantics. Unfortunately, merging XML data is rather complicated and the result is not always clear.

This limit is exceeded by RDF (Resource Description Framework), which is a formal language for describing structured information. RDF is often considered the basic representation format for developing Semantic Web. In contrast with XML, the goal of RDF is not only displaying information in a machine-readable format but also exchanging it on the Web while preserving its original meaning.

RDF documents can be used to represent the relationships among resources using la-belled directed graphs. In this case, the nodes are our resources and the relations are the edge of the graph. For example, the fact that CompanyX collects customer David’s personal information can be represented with the following graph:

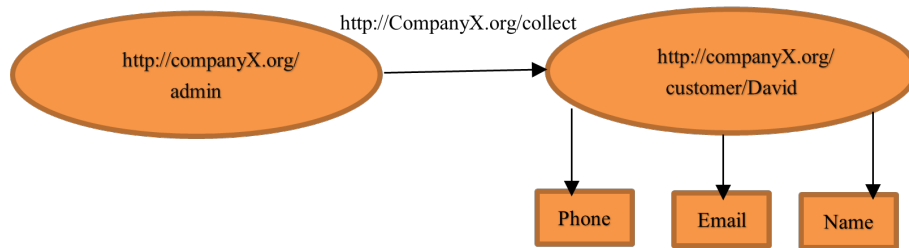


Fig. 5.1: An RDF graph for describing customer data

As it is shown in Figure 5.1, RDF uses a naming system called Universal Resource Identifier (URI). URI is a standard syntax that helps us to simply identify the resources and exchange information on the Web and generally can be assigned to any object that has a clear identity in the privacy policy of the company. In [78], you can find further information about how to create well-formed URIs.

One of the advantages of representing data as RDF graphs is that it is very simple to combine data from multiple sources. This measure is not possible for XML documents because they are encoded in a tree structure and a simple union of two tree structures is not a tree anymore. In the previous example, we represented phone, email, and name as data values or literals. Literals are reserved names for RDF resources of a certain datatype and their values are usually represented as a string like “+417955555”, “david@gmail.com” and “David”.

When companies want to express application-independent knowledge on a given domain, it is possible to use RDF Schema. RDF Schema can be used to express schema or terminological knowledge. A well-known example of RDF Schema is Schema.org a vocabulary that is used by search engines and web applications to empower their user experience. RDF Schema has formal semantics; therefore, we can use RDF reasoners for inferring implicit knowledge from the knowledge that is stated explicitly. For example, by using the clothing materials ontology we can infer that a Lycra blouse is done with synthetic fibre.

5.4. Semantic Web Technologies in Digital Fashion

Now that we know how knowledge can be represented using Semantic Web Technologies, in the next section we present some examples of well-known fashion companies that are currently using Semantic Web technologies to improve their sales and personalization approach.

5.4.2 Related Works Using Semantic Web Technologies in Digital Fashion

As far as we know, there is no previous research using Semantic Web Technologies (SWT) for monitoring privacy policies in a digital fashion. However, SWT has been applied in this domain for other purposes. In [131], they show how in the RISED (Re-factoring Imperial Selling Data) project they manage the collected data from customers using Semantic Web Technologies. They created the Imperial Data Ontology (IDO) and developed visualization tools to analyze all the sale data coming from different databases. One practical advantage of their method is that it can be used to answer many queries that could help the company's sales. For example, the sales department can easily ask the system "What are the best-selling colors within a certain time?". Offering over 2,000 brands in 15 different countries, Zalando is one of the most successful Europe's leading online platforms for fashion. Katarina Kari in Zalando's engineering blog [96] explains how they used SWT to improve their customers' service and personalization by:

- Suggesting links to the customers for further browsing;
- Implementing business rules. For instance, if a customer is browsing a particular brand, the system will not suggest a competing brand;
- Understanding the characteristics of attributes. For example, if the customer searches for vegan coats, then the leather coats will not appear in the result list.

More and more companies are going to use Semantic Web Technologies in the next few years. These related works show just a few advantages of using the Semantic Web in the digital fashion domain.

5.5 The Open Digital Rights Language

All the fashion brands using the Internet as a platform to offer their products to their customers must fulfill the General Data Protection Regulation (GDPR) since it was introduced in May 2018. There exist some tools such as Microsoft Trust Centre[14] and TrustArc [25] which can be used manually by companies to help in assessing the GDPR regulations. However, as the number of companies in the digital fashion world grows, the automated compliance-checking approach can ease these processes [131]. First, we need formal language for specifying our policies. In this section, we describe the ODRL information model and its core classes. Moreover, we show how it is possible to use the ODRL language for formalizing some examples of privacy policies used in the digital fashion domain.

5.5.1 The ODRL Information Model

In general, Digital Rights Management (DRM) systems are responsible for describing, layering, analyzing, trading, and monitoring the rights over digital or physical assets on the Web of Data [85]. Right Expression Languages (REL) is a fundamental part of DRM system, which is machine-readable language, used to express rights. There are several REL standards such as XrML [108], MPEG 21 [84] and other initiatives, but the most common REL standard is the Open Digital Rights Language (ODRL) [77]. The ODRL Information Model defines a set of core classes and properties for expressing a Policy. A Policy must contain at least one Rule object, that is, one object belonging to one of the Rule subclasses are Permission, Prohibition, or Duty. For example, one policy can describe the permission to use customers' contact details and another one can represent the prohibition of sharing customers' sensitive data. A Policy object must belong to one of the following three policy types: Set, Offer, or Agreement. The Set policy is the default type in which any combination of Rules can be represented. The Offer subclass represents Rules that are being offered by an assigner and normally targets a wider audience. The Agreement subclass represents Rules that are granted from an assigner to an assignee. Normally in

5.5. The Open Digital Rights Language

digital fashion when we use the terms assigner and assignee, we consider the fashion companies as assigners who choose the terms of policies and customers as assignees. When there exists an agreement, it means we must have at least one assigner and one assignee. Most of the privacy policies terms are of type Agreement. For example, the Agreement between a customer and a company about which customer data should be collected by a company. Every Policy has a unique identifier. A Policy regulates the Actions performed on an Asset, which is any physical or digital resource or collection of resources. Examples of Assets are the email or the telephone number of a customer. The Actions are labels that can be specified using Constraints, which are Boolean or logical expressions. Actions can be for example “use”, “share”, “transfer”, or “play”. An Action can be permitted on a given asset whereas another one can be prohibited. In addition, we can consider some constraints for each Action. For example, customers can give permission to the company to collect their contact details, but the company can contact the customers only via email. A Policy involves some Parties, which are a person, a collection of people, an organization, or an agent. For instance, fashion companies are the parties that provide services for other parties like customers. A Rule, which can be a Permission, a Prohibition, or a Duty, can be constrained by a condition, if the condition holds the Rule becomes in force. In particular: Permission is used for allowing an Action, when all refinements are satisfied, to be exercised on an Asset. A Prohibition disallows an action, with all refinements satisfied, to be exercised on an Asset even if all constraints are satisfied. Finally, a Duty represents the obligation to exercise an action, with all refinements satisfied. For example, on some fashion websites, customers can be obliged to pay (as a Duty) a small amount of money if they want to access VIP services. In the next session, we will analyze some existing policies used in digital fashion and formalize them using the ODRL language in order to express them in a machine-readable format.

5.5.2 Examples of ODRL policies for the fashion domain

In this Section, we will formalize an existing privacy policy, which is taken from Zalando’s website (Zalando Privacy Notice,2020.) by using ODRL. We want to clarify that Zalando

does not use our approach and to the best of our knowledge, neither Zalando nor any fashion company use automated methods for their policy compliance checking processes.

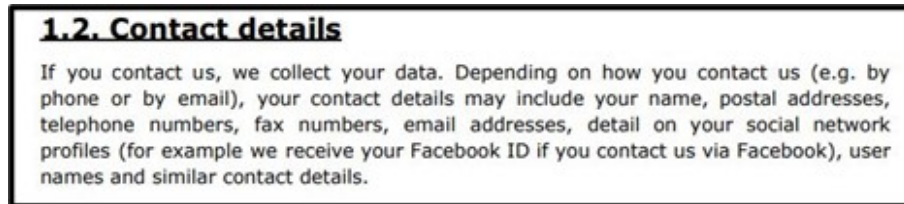


Fig. 5.2: Zalando data collection policy when a customer contacts them

Data collection procedure varies in Zalando based on a wide range of communication between customers and the company. Figure 5.2, represents the policy of the collection of data when a customer contacts Zalando. As it is shown many personal data can be collected depending on how the company is contacted. Such a permission can be represented in ODRL as illustrated in Figure 5.3. To make the formalization of the policy simple, we will formalize only some of the personal information mentioned in the policy reported in Figure 5.2 (i.e. phone number and email). More details on the syntax of ODRL policies can be found on the ODRL website.

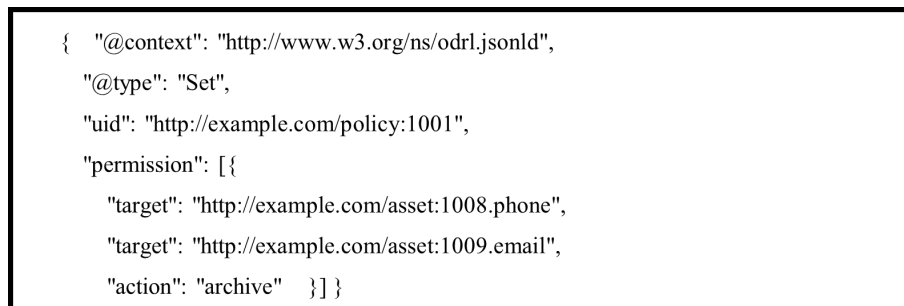


Fig. 5.3: Representation of the permission to archive customer's contact details such as email and phone number in ODRL

Figure 5.4 shows an example of an agreement between CompanyX and user David to use the hometown of the customer. Such a piece of information could significantly improve the sale of the company by getting information about bestselling articles in each city [148]. Figure 5.5 represents a policy that includes two rules one permission and one prohibition on the same resource, "phone number". Customer David gives permission to the company

5.6. Monitoring Norms and Policies

to collect his phone information but on the other hand, the company is prohibited from sharing such data with third parties. In addition, if any conflicts happen between Prohibition and Permission, the conflict property term indicates which one will take precedence. In this example, the conflict property is set to perm which means that the Permission has priority over the Prohibition.

```
{ "@context": "http://www.w3.org/ns/odrl.jsonld",
  "@type": "Agreement",
  "uid": "http://example.com/policy:1002",
  "profile": "http://example.com/odrl:profile:01",
  "permission": [{
    "target": "http://example.com/asset:9898.hometown",
    "assigner": "http://example.com/party:org:CompanyX",
    "assignee": "http://example.com/party:person:CustomerDavid",
```

Fig. 5.4: Representation of 5.2 in ODRL for customer David and CompanyX for the use of David's hometown information. This example highlights an ODRL limitation: it is not possible to specify a template of policies applicable to a set of agents

5.6 Monitoring Norms and Policies

In the preceding sections, we outlined the efforts taken to represent and structure the privacy policy of the companies for the sake of ultimately being accessed and pro-cessed by computers. Moreover, in this section, we study the role of automated policy monitoring. Firstly, we seek to address the importance of monitoring policies and secondly, we discuss the need for some components for automated monitoring. Here we describe a few useful services that can be provided on a set of machine-readable policies:

- Monitoring the compliance of policies in which a person is involved as a debtor. This functionality plays a critical role especially in fashion companies due to their need to collect sensitive customers' data and their need to monitor their employees' behavior toward customers' privacy inside their organization;

```
{  "@context": "http://www.w3.org/ns/odrl.jsonld",
  "@type": "Agreement",
  "uid": "http://example.com/policy:10003",
  "profile": "http://example.com/odrl:profile:08",
  "conflict": "perm",
  "permission": [{
    "target": "http://example.com/phone",
    "action": "collect",
    "assigner": "http://example.com/party:org:CompanyX",
    "assignee": "http://example.com/party:person:CustomerDavid",  }],
  "prohibition": [{
    "target": "http://example.com/phone",
    "action": "share",
    "assigner": "http://example.com/party:org:CompanyX",
    "assignee": "http://example.com/party:person:CustomerDavid",  }]
```

Fig. 5.5: Formalization of one permission and one prohibition in a policy agreement between CompanyX and Customer David. It shows that CompanyX is allowed to collect David's phone number, but it is prohibited to share this information with third-party companies.

- Giving flexibility and confidence to customers by providing a monitoring platform that they can use to see whether their privacy policies are violated or not. For instance, a customer can attach to one picture the prohibition to publish it on a public platform for advertisement and would like to monitor if the actions performed on the picture are compliant with this prohibition [120];
- Searching accurately the resources and the possible actions that can be performed on them. For instance, we assume that the company offered some services to collect some personal data about customers' interests to explore more effective personalization [118].

In order to monitor policies automatically we need the following components:

1. A machine-readable data structure of the actions performed or planned by companies or customers.
2. A mechanism for monitoring the status of policies to check whether there are active

5.7. Conclusion

or not. For example, as mentioned in the previous section, some policies can contain constraints, and as soon as they become satisfied the policies' status will be changed.

3. A mechanism for controlling if a given action (realized on a given resource by a certain agent) is compliant with the set of active policies.
4. A mechanism for automatically computing if the active policies such as obligations or prohibitions are violated or fulfilled.

In the literature, there are two interesting proposals for extending ODRL with operational semantics for monitoring norms and policies automatically. In [28] the model of policies proposed in ODRL has been extended in order to make it possible to express its operational semantics. The authors put in evidence some properties of the policies that are relevant for their life-cycle, in particular, their deadline and their activation condition.

In another interesting paper M. De Vos et al.[44] proposes an ODRL profile that can capture the semantics of both business policies and regularity requirements. They use Answer Set Programming for the policy compliance checking with the possibility of reporting the problem in case compliance is not achieved.

5.7 Conclusion

In this paper, we investigated how we can use ODRL and Semantic Web technologies to express privacy policies of the fashion brands in the Web of Data. We stressed on the transformation of policies from natural language to machine-readable format using RDF. We used ODRL as a policy language for expressing the action that should or should not be performed on the resources. Finally, we have highlighted the literature for monitoring the policies automatically. Our investigations into this area are still ongoing and, in our future, works we plan to investigate the use of the OWL Web Ontology Language for expressing actions performed by the agent and infer their implications by using OWL reasoning.

6

Conclusion

In conclusion, this thesis, we mainly propose a new model of norms that offers a flexible and robust approach to regulating actions with temporal constraints. In addition, we introduced a methodology to help norms design using our model. The relevance of this research comes from the increasing generality of multi-agent systems in our digitally interconnected societies, where agents, whether software or human, interact within complex networks governed by norms, contracts, and agreements. The interdisciplinary nature of this field uses artificial intelligence, communication sciences, digital law, philosophy, and economics, reflecting the diverse facets of normative multi-agent systems.

The primary objectives of this research were threefold. Firstly, the thesis aimed to propose an application-independent norm model capable of overcoming expressivity limitations in existing models. The new proposed model, named T-Norm, introduced temporal constraints and operational semantics to regulate actions within specific temporal intervals using semantic web technologies. Secondly, the thesis introduced a methodology for formalizing existing norms, utilizing the proposed model and complementary existing models. This methodology aimed to facilitate norm designers in their task of expressing and monitoring deontic relations among agents. An important characteristic of the T-Norm model is its ability to specify norms that generate deontic relations at runtime. Notably, it does not rely on predefined deontic types such as obligations or prohibitions. Thirdly, the research focused on the practical application of norm models, specifically addressing the formalization of privacy policies in the digital fashion domain.

The structural overview of the thesis unfolded across six chapters, presenting four papers published (and accepted) in international workshops, conferences, and journals. The T-

NORM model, introduced in Chapter 2, significantly contributed by providing a flexible and robust approach to regulating actions with temporal constraints. This model, leveraging semantic web technologies, offered expressiveness and reasoning capabilities critical for norm monitoring. Chapter 3 extended the research by introducing a methodology for norm formalization, emphasizing machine-readable norms and their benefits. Chapter 4 expanded on these contributions, refining the methodology and introducing a structured approach for defining a rich set of norm types and also representing the normative power and the notion of *Exceptions* in NorMAS.

The application of norm models in real-world scenarios was explored in Chapter 5, addressing privacy concerns in the digital fashion domain. This chapter proposed using semantic web technologies and the Open Digital Rights Language (ODRL) for formalizing machine-readable privacy policies. It underscored the challenges in customer privacy within the digital fashion landscape and outlined how policy formalization could bridge the trust gap between customers and companies.

6.1 Research Contribution

The interdisciplinary nature of normative multi-agent systems was thoroughly investigated, and the proposed T-NORM model, methodology, and practical applications contribute to the growing body of knowledge in this field. As our societies continue to evolve in the digital era, the study of norms within multi-agent systems remains crucial, shaping the landscape of interactions, policies, and regulations governing our interconnected world.

The primary contributions of this article are the development of the T-Norm model and its operational semantics. This model provides a flexible and robust framework for regulating actions with temporal constraints, enabling norm designers to specify and monitor the fulfillment or violation of norms effectively. The use of Semantic Web technologies significantly enhances the model's expressiveness and reasoning capabilities. This research, presented at the 25th International Workshop on Coordination, Organizations, Institutions,

6.1. Research Contribution

Norms, and Ethics for Governance of Multi-Agent Systems (COINE) in 2021, represents a significant advancement in the field.

The proposed model offers several key features that enhance its effectiveness in regulating digital activities. Firstly, it incorporates **temporal constraints**, which allow it to monitor and regulate actions within specific time frames. This is particularly useful in scenarios such as enforcing payment deadlines in e-commerce transactions. The model's ability to handle time-sensitive norms distinguishes it from other approaches.

Secondly, the model utilizes **Semantic Web Technologies**, specifically the Web Ontology Language (OWL), to formalize norms. This enables automatic reasoning about these norms, and, combined with production rules, provides a robust formal semantics for monitoring compliance. The integration of OWL enhances the model's ability to perform sophisticated reasoning tasks, crucial for ensuring that norms are upheld correctly.

Another significant feature is its **expressiveness**. Instead of categorizing norms as obligations or prohibitions, the T-Norm model allows for the detailed expression of the sequence of events leading to the fulfillment or violation of norms. This flexibility allows for a broad spectrum of normative relations, making the model highly adaptable to different scenarios.

The **operational semantics** of the T-Norm model is well-defined by combining OWL reasoning with forward chaining. This combination allows for the automatic monitoring of norms to ensure that any violations are promptly detected and addressed. This operational capability is vital for maintaining the integrity of regulated actions.

Additionally, by converting norms into a machine-readable format, the model facilitates automated analysis and querying. This is especially important in domains such as privacy policy enforcement, where it is critical to monitor compliance with regulations concerning sensitive data, like customers' personal information.

By proposing the T-Norm model, we addressed such questions: "*RQ1. What are the limitations of existing models in expressing and formalizing norms in Multi-agent Systems?*" and "*RQ2. How can an application-independent model of norms be designed to address the expressivity limitations of current models?*".

6.1. Research Contribution

As a second contribution to this research, we suggested a methodology consisting of several essential elements. Initially, it encourages the norm designers to use ontologies as the foundational framework for modeling norms. These formal ontologies establish the basis for defining activation conditions, regulated actions, and constraints related to norms. Specifically, adopting the W3C Web Ontology Language (OWL) is recommended due to its suitability in this domain.

Furthermore, the methodology stresses the importance of identifying whether a norm includes activation conditions or state of affairs. This distinction clarifies the events or circumstances that trigger the norm's applicability and ensures that norms are applied appropriately based on specific triggers (either activation condition or state of affairs).

Moreover, the methodology explores the detailed representation of obligations and prohibitions within normative frameworks. By distinguishing between actions that agents are required to perform and those they are prohibited from performing, the model provides clarity and precision in defining normative guidelines.

Lastly, the methodology addresses both specific and general deontic relations that may arise in Multi-agent Systems. Specific relations involve scenarios where the debtor is known, while general relations extend to a broader set of agents, potentially resulting in multiple violations or fulfillments. This inherent flexibility enables the model to adapt to various scenarios and regulatory requirements, enhancing its practicality and suitability in real-world settings.

By integrating these core components, the proposed methodology establishes a robust framework for formalizing and monitoring norms within Multi-agent Systems. Leveraging semantic web technologies and embracing a comprehensive understanding of normative dynamics, this approach facilitates improved compliance, comprehension, and effectiveness in navigating the complexities of digital interactions. As a result, we addressed the following research questions in this research: "*RQ3. What methodology can be devised for effectively using the proposed normative model and integrating it with complementary models to formalize existing norms?*" and "*RQ4. What challenges arise in the non-trivial task of formalizing diverse norms, and how can these challenges be addressed?*".

6.1. Research Contribution

Chapter 4 builds upon the methodology presented in Chapter 3, introducing significant extensions and enhancements. One major contribution of this chapter is the introduction of a rich set of norm types that can be used to study the expressive power of different formal models of norms and to compare them.

Additionally, Chapter 4 introduces two new concepts that further improve our methodology. The first concept addresses norms that have normative power among agents in Multi-agent Systems, as detailed in Section 4.6.6. This concept is crucial for understanding and modeling the dynamics of power relations within these systems. The second concept focuses on the formalization of exceptions and permissions, as explained in Section 4.6.7. This addition provides a more comprehensive framework for capturing the nuances of normative behaviors, including those that deviate from the standard rules.

These contributions collectively enhance the robustness and applicability of our methodology, offering a more detailed and flexible approach to these research questions: "*RQ5. Can a comprehensive list of various types of norms be created to understand which models are suitable for their formalization?*" and "*RQ6. How feasible is it to translate the formalization of a norm from one model to another, and under what conditions is such translation possible?*".

In addition to the primary objectives of my Ph.D. project, we explored the formalization of human-readable policies in the digital fashion domain into machine-readable formats. This work is presented in Chapter 5 and it discusses the increasing use of digital technologies in fashion communication and e-commerce and the privacy concerns that accompany this shift.

The chapter discusses the rapid growth of digital fashion communication and e-commerce, particularly accelerated by the COVID-19 pandemic. The global fashion industry, which plays a significant role in the economy, is experiencing a digital transformation that brings with it new privacy concerns for both companies and customers. This transformation underscores the need for robust privacy policies to protect sensitive customer data.

The chapter also proposes the use of existing policy expression languages, such as ODRL (Open Digital Rights Language), and Semantic Web Technologies to create machine-

readable privacy policies. This approach aims to facilitate better compliance and understanding of data policies, making them more accessible and actionable.

Furthermore, the importance of policy monitoring is emphasized. The chapter outlines the essential components required for a system to achieve automatic policy monitoring, ensuring that privacy policies are adhered to consistently. It also references related works that provide solutions to the challenges of policy monitoring in the digital fashion domain.

This contribution highlights the potential for formalizing privacy policies to address the privacy challenges in the digital fashion industry and ultimately fostering greater trust and transparency between companies and customers. This work not only advances the field of Multi-agent Systems but also demonstrates the practical application of theoretical models in real-world scenarios. In this chapter we addressed this research question: "*RQ7. How can norms and policy models be practically applied in real-world scenarios, with a focus on privacy policies?*".

6.2 Research Limitations

In addition to the limitations addressed in our papers, this section will discuss more limitations in detail and highlighting important considerations for our future work.

There exists a limited scope of evaluation in the fourth paper (Chapter 4). As it is mentioned in the paper, its contributions and findings have not yet been fully validated by the academic community.

Moreover, The T-NORM model has been tested in a specific context, but there may be limitations in case of generalize it to other contexts or domains. The thesis does not mention extensive testing across diverse scenarios or environments, which is essential to establish the robustness of the proposed model.

The research primarily targets the Normative Multi-agent Systems (NorMAS) and Human-Computer Interaction (HCI) communities. While this focus allows for depth, it might limit

6.2. Research Limitations

the applicability and relevance of the findings to other related fields in computer science and artificial intelligence.

Although the thesis presents theoretical models and methodologies, there is less emphasis on the practical implementation and real-world application of these models. The gap between theoretical formalization and practical usability by norm designers is acknowledged, however, the solutions provided might not fully bridge this gap.

The complexity of the proposed models and methodologies, such as the T-NORM model and the formalization of norms using Semantic Web Technologies may pose a challenge for norm designers who may not have specialized in Semantic Web technologies in these areas.

Regarding privacy Concerns in the Digital Fashion domain, we proposed solutions using Semantic Web Technologies and ODRL. However, it does not provide empirical evidence or case studies demonstrating the effectiveness of these solutions in improving customer trust and privacy protection. Therefore, there is a need for more practical evidence to evaluate the success of this approach.

Another existing challenge in the the T-Norm model is that it operates at such a low logical level that reasoning on the norms themselves becomes challenging. Although the use of OWL allows agents to reason about the contents of norms, such as identifying when a general condition applies to a specific event, the model does not enable agents to reason about the structural relationships between norms. Specifically, T-Norm does not provide a mechanism for detecting conflicts, such as when an obligation in one norm contradicts a prohibition in another, which limits its ability to handle complex normative systems.

Overall, the thesis presents innovative models and methodologies with a strong theoretical foundation, but it faces challenges in validation, practical application, generalizability, and ease of use. Addressing these limitations in future research could enhance the impact and applicability of the proposed contributions.

6.3 Future Work

To address the limitations identified in this thesis, future research should focus on several key areas. First, to enhance the validity of the findings presented in the fourth paper (Chapter 4), it is essential to conduct further evaluations and seek additional peer-reviewed validation. Collaborating with a broader range of academic experts and engaging in more rigorous review processes will help ensure the reliability and acceptance of the research contributions.

Next, the T-NORM model needs to be tested across a wider variety of contexts and domains to determine its generalizability and robustness. Future research should involve applying the model in different scenarios, such as various types of multi-agent systems and real-world environments. This comprehensive testing will help identify any limitations or necessary adjustments to make the model more universally applicable.

Expanding the scope of research beyond the Normative Multi-agent Systems (NorMAS) and Human-Computer Interaction (HCI) communities will also be crucial. Exploring the relevance and applicability of the proposed models and methodologies in other fields within computer science and artificial intelligence will broaden their impact. Interdisciplinary collaboration can uncover new insights and applications that were not initially considered.

Furthermore, there needs to put a stronger emphasis on the practical implementation and usability of the theoretical models. Developing user-friendly tools and frameworks that norm designers can easily adopt, even if they are not experts in Semantic Web Technologies, will bridge the gap between theory and practice. Conducting workshops, training sessions, and creating detailed documentation can also facilitate this transition.

In addressing privacy concerns in the digital fashion domain, future research should focus on providing empirical evidence and conducting case studies. Implementing the proposed solutions using Semantic Web Technologies and ODRL in real-world scenarios will help assess their effectiveness in improving customer trust and privacy protection. Collecting and analyzing data from these implementations will provide valuable feedback for refining

6.3. Future Work

the approaches.

Overall, by addressing these areas, future research can significantly enhance the impact, applicability, and practicality of the innovative models and methodologies presented in this thesis. This will not only validate the theoretical contributions but also ensure their relevance and usefulness in real-world applications.

Moreover, in our future research, we aim to enhance our methodology by continuing the formalization of norms and exploring the potential inclusion of additional types in our type selection. Specifically, we intend to focus on the formalization of sanctions for norms violations and the formalization of constitutive norms.

Additionally, our future work will involve examining the applicability of the proposed methodology to various norm models, particularly those introduced in the legal field, where the "AI and Law" community has presented interesting legal ontologies and legal rule languages[115, 15, 126, 16]. Furthermore, we aim to explore ways to extend the T-Norm model to formalize norms activated by states of affairs.

References

- [1] General data protection regulation, gdpr overview. visited on 2021-02-12. URL: <https://www.microsoft.com/en-us/trust-center/privacy/gdpr-overview>. 132
- [2] Tommy hilfiger introduces tommy jeans xplore smart clothes. visited on 2021-02-12. URL: <https://www.businessinsider.com/tommy-hilfiger-smart-clothes-rewards-2018-7?r=US&IR=T>. 132
- [3] Zalando privacy notice. visited on 2021-02-12. URL: [https://mosaic01.ztat.net/cnt/privacy-page/pdf/Zalando_Privacy_Notice_\(English\).pdf](https://mosaic01.ztat.net/cnt/privacy-page/pdf/Zalando_Privacy_Notice_(English).pdf). 132
- [4] *4th IEEE International Workshop on Policies for Distributed Systems and Networks (POLICY 2003)*, 4-6 June 2003, Lake Como, Italy. IEEE Computer Society, 2003. URL: <http://ieeexplore.ieee.org/xpl/mostRecentIssue.jsp?punumber=8577>.
- [5] *9th IEEE International Workshop on Policies for Distributed Systems and Networks (POLICY 2008)*, 2-4 June 2008, Palisades, New York, USA. IEEE Computer Society, 2008. URL: <http://ieeexplore.ieee.org/xpl/mostRecentIssue.jsp?punumber=4556560>.
- [6] 2017 global mobile consumer survey, 2017. visited on 2021-02-12. URL: <https://www2.deloitte.com/content/dam/Deloitte/us/Documents/technology-media-telecommunications/us-tmt-2017-global-mobile-consumer-survey-executive-summary.pdf>. 132
- [7] Gonzalo A. and Harreis H. Fashion's digital transformation: Now or never, 2020. visited on 2021-02-12. URL: <https://www.mckinsey.com/industries/retail/our-insights/fashions-digital-transformation-now-or-never>. 131

References

- [8] H. Abelson and Creative Commons (Organization). *CcREL: The Creative Commons Rights Expression Language*. 2008. URL: <https://books.google.it/books?id=RSCZAQAACAAJ>. 26
- [9] Harith Alani and Lalana Kagal et al., editors. *ISWC 2013 - 12th International Semantic Web Conference, Sydney, NSW, Australia, October 21-25, 2013, Proceedings, Part I*, volume 8218 of *Lecture Notes in Computer Science*. Springer, 2013. doi:10.1007/978-3-642-41335-3. 165
- [10] Rami Alnaqeib, Fahad H. Alshammari, M. A. Zaidan, A. A. Zaidan, B. B. Zaidan, and Zubaidah M. Hazza. An overview: Extensible markup language technology, 2010. arXiv:1006.4565. 135
- [11] Sergio Álvarez-Napagao, Huib Aldewereld, Javier Vázquez-Salceda, and Frank Dignum. Normative monitoring: Semantics and implementation. In Marina De Vos et al., editors, *COIN@AAMAS 2010, Toronto, Canada, May 2010, COIN@MALLOW 2010, Lyon, France, August 2010, Revised Selected Papers*, volume 6541 of *LNCS*, pages 321–336. Springer, 2010. doi:10.1007/978-3-642-21268-0_18. 44, 58, 61, 68, 69, 93, 103
- [12] Giulia Andrighetto, Guido Governatori, Pablo Noriega, and Leendert W. N. van der Torre, editors. *Normative Multi-Agent Systems*, volume 4 of *Dagstuhl Follow-Ups*. Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 2013. URL: <http://drops.dagstuhl.de/opus/portals/dfu/index.php?semnr=13003>. 11, 66, 90, 96
- [13] Grigoris Antoniou, Paul Groth, Frank van Harmelen, and Rinke Hoekstra. *A Semantic Web Primer*. The MIT Press, 2012. 19, 20
- [14] Grigoris Antoniou and Frank van Harmelen. *Web Ontology Language: OWL*, pages 91–110. Springer Berlin Heidelberg, Berlin, Heidelberg, 2009. doi:10.1007/978-3-540-92673-3_4. 20
- [15] Tara Athan, Harold Boley, Guido Governatori, Monica Palmirani, Adrian Paschke, and Adam Wyner. Oasis legalruleml. In *Proceedings of the Fourteenth International Conference on Artificial Intelligence and Law, ICAIL '13*, page 3–12, New York,

References

- NY, USA, 2013. Association for Computing Machinery. doi:10.1145/2514601.2514603. 93, 153
- [16] Tara Athan, Guido Governatori, Monica Palmirani, Adrian Paschke, and Adam Z. Wyner. Legalruleml: Design principles and foundations. In Wolfgang Faber and Adrian Paschke, editors, *Reasoning Web. Web Logic Rules - 11th International Summer School 2015, Berlin, Germany, July 31 - August 4, 2015, Tutorial Lectures*, volume 9203 of *Lecture Notes in Computer Science*, pages 151–188. Springer, 2015. doi:10.1007/978-3-319-21768-0_6. 15, 153
- [17] P. G. Balaji and D. Srinivasan. *An Introduction to Multi-Agent Systems*, pages 1–27. Springer Berlin Heidelberg, Berlin, Heidelberg, 2010. doi:10.1007/978-3-642-14435-6_1. 10
- [18] Zenon Bankowski. Norms, legal. 1998. URL: <https://www.rep.routledge.com/articles/thematic/norms-legal/v-1>, doi:10.4324/9780415249126-T021-1.
- [19] Nick Bassiliades, Georg Gottlob, Fariba Sadri, Adrian Paschke, and Dumitru Roman, editors. *Rule Technologies: Foundations, Tools, and Applications - 9th International Symposium, RuleML 2015, Berlin, Germany, August 2-5, 2015, Proceedings*, volume 9202 of *Lecture Notes in Computer Science*. Springer, 2015. doi:10.1007/978-3-319-21542-6. 175
- [20] Stefan Becker, Benjamin Hück, Katharina Naujokat, Arne Fritjof Schmeiser, and Andreas Kasten. ODRL 2.0 revisited. In Matthias Horbach, editor, *GI-Jahrestagung*, volume 220 of *LNI*, pages 3081–3095. GI, 2013. URL: <http://dblp.uni-trier.de/db/conf/gi/gi2013.html#BeckerHNSK13>.
- [21] Francesco Belardinelli and Estefania Argente, editors. *Multi-Agent Systems and Agreement Technologies - 15th European Conference, EUMAS 2017, and 5th International Conference, AT 2017, Évry, France, December 14-15, 2017, Revised Selected Papers*, volume 10767 of *Lecture Notes in Computer Science*. Springer, 2018. doi:10.1007/978-3-030-01713-2. 162

References

- [22] Christoph Benzmüller, Xavier Parent, and Leendert W. N. van der Torre. Designing normative theories for ethical and legal reasoning: Logikey framework, methodology, and tool support. *Artificial Intelligence*, 287:103348, 2020. URL: <https://www.sciencedirect.com/science/article/pii/S0004370219301110>, doi:10.1016/j.artint.2020.103348. 68, 90
- [23] Berners-Lee and Rudolph. The semantic web a new form of web content that is meaningful to computers will unleash a revolution of new possibilities., 2009. visited on 2021-02-12. URL: <https://www.scientificamerican.com/article/the-semantic-web/>. 135
- [24] Tim Berners-Lee, James Hendler, and Ora Lassila. The semantic web. *Scientific American*, 284(5):34–43, 2001. Accessed 20 Nov. 2023. URL: <http://www.jstor.org/stable/26059207>. 18
- [25] Guido Boella and Leendert van der Torre. Regulative and constitutive norms in normative multiagent systems. In *Proceedings of the Ninth International Conference on Principles of Knowledge Representation and Reasoning, KR’04*, page 255–265, California, 2004. AAAI Press. 12, 122
- [26] Guido Boella and Leendert van der Torre. Permission and authorization in normative multiagent systems. In *Proceedings of the 10th International Conference on Artificial Intelligence and Law, ICAIL ’05*, page 236–237, New York, NY, USA, 2005. Association for Computing Machinery. doi:10.1145/1165485.1165526. 122
- [27] Guido Boella, Leon van der Torre, and Harko Verhagen. Introduction to normative multiagent systems. *Computational & Mathematical Organization Theory*, 12, 01 2006. doi:10.1007/s10588-006-9537-7. 15, 17
- [28] Kurt D. Bollacker, Natalia Díaz-Rodríguez, Xian Li, and Stitch. Beyond clothing ontologies : Modeling fashion with subjective influence networks. 2016. URL: <https://api.semanticscholar.org/CorpusID:53135605>. 134, 143
- [29] P. Bonatti and D. Olmedilla. Driving and monitoring provisional trust negotiation

References

- with metapolicies. In *Sixth IEEE International Workshop on Policies for Distributed Systems and Networks (POLICY'05)*, pages 14–23, 2005. doi:10.1109/POLICY.2005.13. 66, 91
- [30] Piero A. Bonatti and Daniel Olmedilla. *Rule-Based Policy Representation and Reasoning for the Semantic Web*, pages 240–268. Springer Berlin Heidelberg, Berlin, Heidelberg, 2007. doi:10.1007/978-3-540-74615-7_4. 66, 91
- [31] Ronald Brachman and Hector Levesque. *Knowledge Representation and Reasoning*. Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, 2004.
- [32] Ronald Brachman and Hector Levesque. *Knowledge Representation and Reasoning*. Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, 2004. 53
- [33] Jeffrey M. Bradshaw, Andrzej Uszok, Maggie Breedy, Larry Bunch, Thomas C. Eskridge, Paul J. Feltovich, Matthew Johnson, James Lott, and Micael Vignati. The KAoS Policy Services Framework. Eighth Cyber Security and Information Intelligence Research Workshop (CSIIRW 2013). Oak Ridge, TN: Oak Ridge National Labs, 2013. 32, 66
- [34] Arnaud Braud, Gaël Fromentoux, Benoit Radier, and Olivier Le Grand. The road to european digital sovereignty with gaia-x and idsa. *IEEE Network*, 35(2):4–5, 2021. doi:10.1109/MNET.2021.9387709. 34
- [35] Jorge Cardoso and Amit Sheth. *The Semantic Web and Its Applications*, pages 3–33. Springer US, Boston, MA, 2006. doi:10.1007/978-0-387-34685-4_1. 17
- [36] Ronald P. Carver. Is reading rate constant or flexible? *Reading Research Quarterly*, 18(2):190–215, 1983. URL: <http://www.jstor.org/stable/747517>. 132, 134
- [37] Amit Chopra, Leendert van der Torre, Harko Verhagen, and Serena Villata, editors. *Handbook of Normative Multiagent Systems*. College Publications, August 2018. 66, 90, 96
- [38] Adrian Paschke Christian de Sainte Marie, Gary Hallmark. RIF Production Rule Dialect (Second Edition) W3C Recommendation 5 February 2013. 2013.

References

- URL: https://www.w3.org/TR/rif-prd/#Operational_semantics_of_a_production_rule_system.
- [39] Luca Costabello, Serena Villata, and Fabien L. Gandon. Context-aware access control for rdf graph stores. In *European Conference on Artificial Intelligence*, 2012. URL: <https://api.semanticscholar.org/CorpusID:4979708>. 33
- [40] N. Criado, E. Argente, and V. Botti. Open issues for normative multi-agent systems. *AI Commun.*, 24(3):233–264, aug 2011. 25
- [41] Karen da Silva Figueiredo, Viviane Torres da Silva, and Christiano de Oliveira Braga. *Modeling Norms in Multi-agent Systems with NormML*, pages 39–57. Springer Berlin Heidelberg, Berlin, Heidelberg, 2011. doi:10.1007/978-3-642-21268-0_3.
- [42] Alaa Daoud. Semantic web environments for multi-agent systems: Enabling agents to use web of things via semantic web. *arXiv preprint arXiv:2003.02054*, 2020. 22
- [43] Mehdi Dastani, Davide Grossi, John-Jules Ch. Meyer, and Nick Tinnemeier. Normative multi-agent programs and their logics. In John-Jules Ch. Meyer and Jan Broersen, editors, *Knowledge Representation for Agents and Multi-Agent Systems*, pages 16–31, Berlin, Heidelberg, 2009. Springer Berlin Heidelberg. 16
- [44] Marina De Vos, Sabrina Kirrane, Julian Padget, and Ken Satoh. ODRL Policy Modelling and Compliance Checking. In Paul Fodor, Marco Montali, Diego Calvanese, and Dumitru Roman, editors, *Rules and Reasoning*, pages 36–51, Cham, 2019. Springer International Publishing. 41, 59, 61, 67, 92, 143
- [45] Robert Demolombe and Vincent Louis. Norms, institutional power and roles: Towards a logical framework. In Floriana Esposito, Zbigniew W. Ras, Donato Malerba, and Giovanni Semeraro, editors, *Foundations of Intelligent Systems, 16th International Symposium, ISMIS 2006, Bari, Italy, September 27-29, 2006, Proceedings*, volume 4203 of *Lecture Notes in Computer Science*, pages 514–523, Berlin, Heidelberg, 2006. Springer. doi:10.1007/11875604_58. 101
- [46] Frank Dignum, Jan Broersen, Virginia Dignum, and John-Jules Meyer. Meeting

References

- the deadline: Why, when and how. In Michael G. Hinchey, James L. Rash, Walter F. Truszkowski, and Christopher A. Rouff, editors, *Formal Approaches to Agent-Based Systems*, pages 30–40, Berlin, Heidelberg, 2005. Springer Berlin Heidelberg. 96, 99, 100
- [47] Virginia Dignum, Javier Vazquez-Salceda, and Frank Dignum. A model of almost everything: Norms, structure and ontologies in agent organizations. In *Proceedings of the Third International Joint Conference on Autonomous Agents and Multiagent Systems - Volume 3*, AAMAS '04, page 1498–1499, USA, 2004. IEEE Computer Society. 11
- [48] M. Esteva, B. Rosell, J.A. Rodriguez-Aguilar, and J.Ll. Arcos. Ameli: an agent-based middleware for electronic institutions. In *Proceedings of the Third International Joint Conference on Autonomous Agents and Multiagent Systems, 2004. AAMAS 2004.*, pages 236–243, 2004. 16
- [49] Beatriz Esteves, Harshvardhan J. Pandit, and Víctor Rodríguez-Doncel. Odrl profile for expressing consent through granular access control policies in solid. In *2021 IEEE European Symposium on Security and Privacy Workshops*, pages 298–306, 2021. doi:10.1109/EuroSPW54576.2021.00038. 30
- [50] Beatriz Esteves, Víctor Rodríguez-Doncel, Harshvardhan J. Pandit, Nicolas Mondada, and Pat McBennett. Using the odrl profile for access control for solid pod resource governance. In Paul Groth, Anisa Rula, Jodi Schneider, Ilaria Tiddi, Elena Simperl, Panos Alexopoulos, Rinke Hoekstra, Mehwish Alam, Anastasia Dimou, and Minna Tamper, editors, *The Semantic Web: ESWC 2022 Satellite Events*, pages 16–20, Cham, 2022. Springer International Publishing. 30
- [51] Anna Fensel, Amrapali Zaveri, Sebastian Hellmann, and Tassilo Pellegrini, editors. *Proceedings of the 12th International Conference on Semantic Systems, SEMANTICS 2016, Leipzig, Germany, September 12-15, 2016*. ACM, 2016. URL: <http://dl.acm.org/citation.cfm?id=2993318>. 163
- [52] T. Finin, A. Joshi, L. Kagal, J. Niu, R. Sandhu, W. Winsborough, and B. Thuraisingham. Rowlbac: representing role based access control in owl. In *Proceedings of the*

- 13th ACM Symposium on Access Control Models and Technologies, SACMAT '08*, page 73–82, New York, NY, USA, 2008. Association for Computing Machinery. doi:10.1145/1377836.1377849. 33
- [53] Charles Lanny Forgy. *On the Efficient Implementation of Production Systems*. PhD thesis, Pittsburgh, PA, USA, 1979. AAI7919143.
- [54] Nicoletta Fornara. Specifying and Monitoring Obligations in Open Multiagent Systems using Semantic Web Technology. In *Semantic Agent Systems: Foundations and Applications*, volume 344 of *Studies in Computational Intelligence*, chapter 2, pages 25–46. Springer-Verlag, 2011. 22, 31, 45, 134
- [55] Nicoletta Fornara and Marco Colombetti. Operational specification of a commitment-based agent communication language. In *Proceedings of the First International Joint Conference on Autonomous Agents and Multiagent Systems: Part 2, AAMAS '02*, pages 536–542, New York, NY, USA, 2002. ACM. URL: <http://doi.acm.org/10.1145/544862.544868>, doi:10.1145/544862.544868.
- [56] Nicoletta Fornara and Marco Colombetti. Specifying and enforcing norms in artificial institutions. In Matteo Baldoni, Tran Cao Son, M. Birna van Riemsdijk, and Michael Winikoff, editors, *Declarative Agent Languages and Technologies VI*, pages 1–17, Berlin, Heidelberg, 2009. Springer Berlin Heidelberg. 64, 86
- [57] Nicoletta Fornara and Marco Colombetti. Ontology and time evolution of obligations and prohibitions using semantic web technology. In Matteo Baldoni, Jamal Bentahar, M. Birna van Riemsdijk, and John Lloyd, editors, *Declarative Agent Languages and Technologies VII*, pages 101–118, Berlin, Heidelberg, 2010. Springer Berlin Heidelberg.
- [58] Nicoletta Fornara and Marco Colombetti. Representation and monitoring of commitments and norms using OWL. *AI Commun.*, 23(4):341–356, 2010. 21, 31, 57
- [59] Nicoletta Fornara and Marco Colombetti. Operational semantics of an extension of ODRL able to express obligations. In Belardinelli and Argente [21], pages 172–186. doi:10.1007/978-3-030-01713-2_13.

References

- [60] Nicoletta Fornara and Marco Colombetti. Using semantic web technologies and production rules for reasoning on obligations, permissions, and prohibitions. *AI Commun.*, 32(4):319–334, 2019. doi:10.3233/AIC-190617. 41, 42, 45, 59, 60, 65, 67, 77, 86, 92, 116, 134
- [61] Nicoletta Fornara and Fabio Marfia. Modeling and enforcing access control obligations for SPARQL-DL queries. In Fensel et al. [51], pages 145–152. URL: <http://doi.acm.org/10.1145/2993318.2993337>, doi:10.1145/2993318.2993337. 31
- [62] Nicoletta Fornara, Soheil Roshankish, and Marco Colombetti. A framework for automatic monitoring of norms that regulate time constrained actions. In *Proceedings of the International Workshop on Coordination, Organizations, Institutions, Norms and Ethics for Governance of Multi-Agent Systems (COINE), co-located with AAMAS 2021, 3rd May 2021, London, UK, 2021*. arXiv, 2021. URL: <https://arxiv.org/abs/2105.00200>, doi:10.48550/ARXIV.2105.00200. 65, 67, 68, 86, 87, 92, 102, 118
- [63] Nicoletta Fornara and Marco Sterpetti. An architecture for monitoring norms that combines OWL reasoning and forward chaining over rules. In Emilio M. Sanfilippo et al., editor, *Proceedings of the Joint Ontology Workshops 2021 Episode VII: The Bolzano Summer of Knowledge co-located with the 12th International Conference on Formal Ontology in Information Systems (FOIS 2021), and the 12th International Conference on Biomedical Ontologies (ICBO 2021), Bolzano, Italy, September 11-18, 2021*, volume 2969 of *CEUR Workshop Proceedings*. CEUR-WS.org, 2021. URL: <http://ceur-ws.org/Vol-2969/paper8-DEM0.pdf>. 7, 67, 68, 87, 92, 102
- [64] Nicoletta Fornara and Charalampos Tampitsikas. Semantic technologies for open interaction systems. *Artificial Intelligence Review*, 39(1):63–79, 2013. doi:10.1007/s10462-012-9390-7. 31
- [65] Kobi Gal and Barbara J. Grosz. Multi-Agent Systems: Technical and Ethical Challenges of Functioning in a Mixed Group. *Daedalus*,

References

- 151(2):114–126, 05 2022. URL: https://doi.org/10.1162/daed_a_01904, arXiv:https://direct.mit.edu/daed/article-pdf/151/2/114/2060592/daed_a_01904.pdf, doi:10.1162/daed_a_01904. 24
- [66] Fabien Gandon, Guido Governatori, and Serena Villata. Normative requirements as linked data. In Adam Z. Wyner and Giovanni Casini, editors, *Legal Knowledge and Information Systems - JURIX 2017: The Thirtieth Annual Conference, Luxembourg, 13-15 December 2017*, volume 302 of *Frontiers in Artificial Intelligence and Applications*, pages 1–10, Amsterdam, 2017. IOS Press. doi:10.3233/978-1-61499-838-9-1. 90
- [67] Emilia Garcia, Simon Miles, Michael Luck, and Adriana Giret. Evaluating how agent methodologies support the specification of the normative environment through the development process. *Autonomous Agents and Multi-Agent Systems*, 29(6):1041–1060, nov 2015. doi:10.1007/s10458-014-9275-z.
- [68] A. Garcia-Camino, P. Noriega, and J. A. Rodriguez-Aguilar. Implementing norms in electronic institutions. In *Proceedings of the Fourth International Joint Conference on Autonomous Agents and Multiagent Systems, AAMAS '05*, pages 667–673, New York, NY, USA, 2005. ACM. URL: <http://doi.acm.org/10.1145/1082473.1082575>, doi:10.1145/1082473.1082575. 11, 43, 58, 61, 68, 93
- [69] Teresa Sádaba Garraza. Fashion in the digital environment, 2015. visited on 2021-02-12. URL: https://www.eunsa.es/libro/moda-en-el-entorno-digital_102440/. 131
- [70] J. P. Gibbs. Norms: The problem of definition and classification. *The American Journal of Sociology*, 70(5):586–594, 1965. 11
- [71] Stevan Gostojić, Branko Milosavljević, and Zora Konjović. Ontological model of legal norms for creating and using legislation. *Computer Science and Information Systems*, 10(1):151–171, 2013. doi:10.2298/CSIS110804035G. 32
- [72] Guido Governatori and Renato Iannella. A modelling and reasoning framework for social networks policies. *Enterp. Inf. Syst.*, 5(1):145–167, feb 2011. doi:

References

- 10.1080/17517575.2010.513014. 67, 92
- [73] Guido Governatori, Francesco Olivieri, Antonino Rotolo, and Simone Scannapieco. Computing strong and weak permissions in defeasible logic. *Journal of Philosophical Logic*, 42(6):799–829, Dec 2013. doi:10.1007/s10992-013-9295-1. 50, 123
- [74] Guido Governatori and Antonino Rotolo. BIO logical agents: Norms, beliefs, intentions in defeasible logic. *Autonomous Agents and Multi-Agent Systems*, 17(1):36–69, 2008. doi:10.1007/s10458-008-9030-4.
- [75] Guido Governatori and Antonino Rotolo. A conceptually rich model of business process compliance. In *Proceedings of the Seventh Asia-Pacific Conference on Conceptual Modelling - Vol. 110*, pages 3—12. Australian Computer Society, Inc., 2010. 28, 44, 59
- [76] Guido Governatori, Antonino Rotolo, Serena Villata, and Fabien Gandon. One license to compose them all - A deontic logic approach to data licensing on the web of data. In Alani and et al. [9], pages 151–166. doi:10.1007/978-3-642-41335-3_10.
- [77] Susanne Guth. *Rights Expression Languages*, pages 101–112. Springer Berlin Heidelberg, Berlin, Heidelberg, 2003. doi:10.1007/10941270_8. 138
- [78] Pascal Hitzler, Markus Krtzsch, and Sebastian Rudolph. *Foundations of Semantic Web Technologies*. Chapman & Hall/CRC, 1st edition, 2009. 135, 136
- [79] Pascal Hitzler, Markus Krötzsch, and Sebastian Rudolph. *Foundations of Semantic Web Technologies*. Illustrated. CRC Press, Boca Raton, 1st edition, 2010. TK5105.88815 .H57 2010. 19
- [80] Rinke Hoekstra, editor. *Legal Knowledge and Information Systems - JURIX 2014: The Twenty-Seventh Annual Conference, Jagiellonian University, Krakow, Poland, 10-12 December 2014*, volume 271 of *Frontiers in Artificial Intelligence and Applications*. IOS Press, 2014. 173

References

- [81] Wesley Hohfeld. Fundamental legal conceptions as applied in judicial reasoning, 1917. The present discussion, while intended to be intrinsically complete so far as intelligent and convenient perusal is concerned, represents, as originally planned, a continuation of an article which appeared under the same title more than three years ago. It therefore seems desirable to indicate, in very general form, the scope and purpose of the latter. The main divisions were entitled: Legal Conceptions Contrasted with Non-legal Conceptions; Operative Facts Contrasted with Evidential Facts; and Fundamental Jural Relations Contrasted with One Another. The jural relations analyzed and discussed under the last subtitle were, at the outset, grouped in a convenient "scheme of opposites and correlatives"; and it will greatly facilitate the presentation of the matters to be hereafter considered if that scheme be reproduced at the present point:. URL: <http://hdl.handle.net/20.500.13051/3864>. 23
- [82] Ian Horrocks. OWL: A Description Logic Based Ontology Language. In Peter van Beek, editor, *Principles and Practice of Constraint Programming - CP 2005*, pages 5–8, Berlin, Heidelberg, 2005. Springer Berlin Heidelberg. 39
- [83] Jomi Fred Hübner, Jaime Simão 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, 2007. doi:10.1504/IJA0SE.2007.016266. 11
- [84] Burnett I., R Van de Walle, K Hill Rightscom, U Bormans, and F. Pereira. Mpeg-21: goals and achievements. *IEEE Multimedia*, 2003. 138
- [85] Renato Iannella. The open digital rights language: Xml for digital rights management. *Information Security Technical Report*, 9(3):47–55, 2004. URL: <https://www.sciencedirect.com/science/article/pii/S1363412704000317>, doi:10.1016/S1363-4127(04)00031-7. 138
- [86] Renato Iannella, Susanne Guth, Dennis Paehler, and Andreas Kasten. ODRL Version 2.1 Core Model. <https://www.w3.org/community/odrl/model/2.1/>, 2015. Accessed on 15/02/2017. 26

References

- [87] Renato Iannella and Serena Villata. ODRL Information Model 2.2 W3C Recommendation 15 February 2018. <https://www.w3.org/TR/odrl-model/> (last accessed Nov.2020), 2018.
- [88] Prateek Jain, Pascal Hitzler, Krzysztof Janowicz, and Chitra Venkatramani. There’s no money in linked data. 2013.
- [89] Samyar Janatian, Hannes Westermann, Jinzhe Tan, Jaromir Savelka, and Karim Benyekhlef. From text to structure: Using large language models to support the development of legal expert systems, 2023. URL: <https://arxiv.org/abs/2311.04911>, arXiv:2311.04911. 34
- [90] Nicholas R. Jennings, Katia Sycara, and Michael Wooldridge. A Roadmap of Agent Research and Development. *Autonomous Agents and Multi-Agent Systems*, 1(1):7–38, March 1998. doi:10.1023/A:1010090405266. 10
- [91] A. J. I. Jones and M. Sergot. A Formal Characterisation of Institutionalised Power. *Logic Journal of the IGPL*, 4(3):427–443, 06 1996. doi:10.1093/jigpal/4.3.427. 96, 100
- [92] Vicente Julian and Vicente Botti. Multi-agent systems. *Applied Sciences*, 9(7), 2019. URL: <https://www.mdpi.com/2076-3417/9/7/1402>, doi:10.3390/app9071402. 15
- [93] Lalana Kagal. *A Policy-Based Approach to Governing Autonomous Behavior in Distributed Environments*. PhD thesis, University of Maryland Baltimore County, Baltimore MD 21250, September 2004. 66, 91
- [94] Lalana Kagal, Timothy W. Finin, and Anupam Joshi. A policy language for a pervasive computing environment. In *4th IEEE International Workshop on Policies for Distributed Systems and Networks (POLICY 2003)*, 4-6 June 2003, Lake Como, Italy, page 63, 2003. doi:10.1109/POLICY.2003.1206958. 32
- [95] Nadzeya Kalbaska and Lorenzo Cantoni. Digital fashion competences: Market practices and needs. In Rinaldo Rinaldi and Romeo Bandinelli, editors, *Business Models and ICT Technologies for the Fashion Supply Chain*, pages 125–135, Cham,

2019. Springer International Publishing. 131
- [96] Katariina Kari. The art of ontology. visited on 2021-02-12. URL: <https://engineering.zalando.com/posts/2018/03/semantic-web-technologies.html>. 131, 137
- [97] A. Kasten and R. Grimm. Making the semantics of odr1 and urm explicit using web ontologies. In *Virtual Goods*, pages 77–91, 2010. 27
- [98] Sabrina Kirrane, Serena Villata, and Mathieu d’Aquin. Privacy, security and policies: A review of problems and solutions with semantic web technologies. *Semantic Web*, 9(2):153–161, 2018. doi:10.3233/SW-180289. 24, 133
- [99] Nadin Kokciyan and Pinar Yolum. Priguard: A semantic approach to detect privacy violations in online social networks. *IEEE Trans. on Knowl. and Data Eng.*, 28(10):2724–2737, October 2016. doi:10.1109/TKDE.2016.2583425.
- [100] Panos Kudumakis, Thomas Wilmering, Mark Sandler, Victor Rodriguez-Doncel, Laurent Boch, and Jaime Delgado. The challenge: From mpeg intellectual property rights ontologies to smart contracts and blockchains [standards in a nutshell]. *IEEE Signal Processing Magazine*, 37(2):89–95, 2020. doi:10.1109/MSP.2019.2955207. 26
- [101] Ninghui Li. *Discretionary Access Control*, pages 353–356. Springer US, Boston, MA, 2011. doi:10.1007/978-1-4419-5906-5_798. 12
- [102] Fabiola López López, Michael Luck, and Mark d’Inverno. Constraining autonomy through norms. In *Proceedings of the First International Joint Conference on Autonomous Agents and Multiagent Systems: Part 2, AAMAS ’02*, page 674–681, New York, NY, USA, 2002. Association for Computing Machinery. doi:10.1145/544862.544905. 11
- [103] Moamin A. Mahmoud, Mohd Sharifuddin Ahmad, Mohd Zaliman Mohd Yusoff, and Aida Mustapha. A review of norms and normative multiagent systems. *The Scientific World Journal*, 2014:684587, 2014. doi:10.1155/2014/684587. 12

References

- [104] Fabio Marfia, Nicoletta Fornara, and Truc-Vien T. Nguyen. A framework for managing data provider and data consumer semantic obligations for access control. *Unknown Journal*, pages 67–82, Jan 2017. 31
- [105] Aleecia M. McDonald and Lorrie Faith Cranor. The cost of reading privacy policies. *A Journal of Law and Policy for the Information Society*, 2008. 132
- [106] Georgios Meditskos and Nick Bassiliades. Dlejena: A practical forward-chaining OWL 2 RL reasoner combining jena and pellet. *J. Web Sem.*, 8(1):89–94, 2010. doi:10.1016/j.websem.2009.11.001.
- [107] J.-J.Ch. Meyer, Roelf J. Wieringa, and F.P.M. Dignum. *The role of deontic logic in the specification of information systems*, pages 71–115. Kluwer Academic Publishers, Netherlands, 1998. 15
- [108] Microsoft. Xrml elements. visited on 2021-02-12. URL: https://learn.microsoft.com/en-us/previous-versions/windows/desktop/adrms_sdk/xrml-elements. 138
- [109] Javier Morales, Maite Lopez-Sanchez, Juan A. Rodriguez-Aguilar, Michael Wooldridge, and Wamberto Vasconcelos. Automated synthesis of normative systems. In *Proceedings of the 2013 International Conference on Autonomous Agents and Multi-Agent Systems*, AAMAS '13, page 483–490, Richland, SC, 2013. International Foundation for Autonomous Agents and Multiagent Systems. 94
- [110] Javier Morales, Michael Wooldridge, Juan A. Rodríguez-Aguilar, and Maite López-Sánchez. Off-line synthesis of evolutionarily stable normative systems. *Autonomous Agents and Multi-Agent Systems*, 32(5):635–671, sep 2018. doi:10.1007/s10458-018-9390-3. 94
- [111] A. Morris-Martin, M. De Vos, and J. Padget. Norm emergence in multiagent systems: A viewpoint paper. *Autonomous Agents and Multi-Agent Systems*, 33:706–749, 2019. doi:10.1007/s10458-019-09422-0.
- [112] Andreea Morris-Martin, Marina De Vos, Julian A. Padget, and Oliver Ray. Agent-directed runtime norm synthesis. In Noa Agmon, Bo An, Alessandro

- Ricci, and William Yeoh, editors, *Proceedings of the 2023 International Conference on Autonomous Agents and Multiagent Systems, AAMAS 2023, London, United Kingdom, 29 May 2023 - 2 June 2023*, pages 2271–2279. ACM, 2023. URL: <https://dl.acm.org/doi/10.5555/3545946.3598905>, doi: 10.5555/3545946.3598905. 24, 25
- [113] T. Moses. extensible access control markup language (xacml) version 2.0. OASIS Standard, 200502, 2005. 28
- [114] Jakub Moskal and Christopher J. Matheus. Detection of Suspicious Activity Using Different Rule Engines - Comparison of BaseVISor, Jena and Jess Rule Engines. In N.Bassiliades et al. [115], pages 73–80. doi:10.1007/978-3-540-88808-6_10. 15
- [115] N.Bassiliades, G. Governatori, and A. Paschke, editors. *Rule Representation, Interchange and Reasoning on the Web, International Symposium, RuleML 2008, Orlando, FL, USA, October 30-31, 2008. Proceedings*, volume 5321 of *LNCS*. Springer, 2008. doi:10.1007/978-3-540-88808-6. 28, 153, 170
- [116] Truc-Vien T. Nguyen and Nicoletta Fornara. Policy enforcement for the web of data. In *Proceedings of the 2014 Workshop on Intelligent Agents and Technologies for Socially Interconnected Systems, IAT4SIS '14*, page 1–6, New York, NY, USA, 2014. Association for Computing Machinery. doi:10.1145/2655985.2655988. 31
- [117] Truc-Vien T. Nguyen, Nicoletta Fornara, and Fabio Marfia. Automatic policy enforcement on semantic social data. *Multiagent and Grid Systems*, 11(3):121–146, 2015. doi:10.3233/MGS-150233. 31
- [118] Tekila Harley Nobile and Nadzeya Kalbaska. An exploration of personalization in digital communication. insights in fashion. In *HCI in Business, Government and Organizations: 7th International Conference, HCIBGO 2020, Held as Part of the 22nd HCI International Conference, HCII 2020, Copenhagen, Denmark, July 19–24, 2020, Proceedings*, page 456–473, Berlin, Heidelberg, 2020. Springer-Verlag. doi:10.1007/978-3-030-50341-3_35. 142

References

- [119] Donald Nute. Defeasible logic. In Oskar Bartenstein, Ulrich Geske, Markus Hannebauer, and Osamu Yoshie, editors, *Web Knowledge Management and Decision Support*, pages 151–169, Berlin, Heidelberg, 2003. Springer Berlin Heidelberg. 28
- [120] Alessandro Oltramari, Dhivya Piraviperumal, Florian Schaub, Shomir Wilson, Sushain Cherivirala, Thomas B. Norton, N. Cameron Russell, Peter Story, Joel R. Reidenberg, and Norman M. Sadeh. Privonto: A semantic framework for the analysis of privacy policies. *Semantic Web*, 9(2):185–203, 2018. doi:10.3233/SW-170283. 59, 65, 86, 131, 142
- [121] Nir Oren, Michael Luck, and Simon Miles. A model of normative power. In Wiebe van der Hoek, Gal A. Kaminka, Yves Lespérance, Michael Luck, and Sandip Sen, editors, *9th International Conference on Autonomous Agents and Multiagent Systems (AAMAS 2010), Toronto, Canada, May 10-14, 2010, Volume 1-3*, pages 815–822, Richland, SC, 2010. IFAAMAS. URL: <https://dl.acm.org/citation.cfm?id=1838315>. 96, 100, 121
- [122] Michela Ornati and Lorenzo Cantoni. Fashiontouch in e-commerce: An exploratory study of surface haptic interaction experiences. In Fiona Fui-Hoon Nah and Keng Siau, editors, *HCI in Business, Government and Organizations*, pages 493–503, Cham, 2020. Springer International Publishing. 131
- [123] Sylvia L. Osborn. *Role-Based Access Control*, pages 55–70. Springer Berlin Heidelberg, Berlin, Heidelberg, 2007. doi:10.1007/978-3-540-69861-6_5.
- [124] Julian Padget, Emad ElDeen Elakehal, Tingting Li, and Marina De Vos. *InstAL: An Institutional Action Language*, pages 101–124. Springer International Publishing, Cham, 2016. doi:10.1007/978-3-319-33570-4_6. 92
- [125] Julian Padget, Marina De Vos, and Charlie Ann Page. Deontic sensors. In *Proceedings of the Twenty-Seventh International Joint Conference on Artificial Intelligence, IJCAI-18*, pages 475–481. International Joint Conferences on Artificial Intelligence Organization, 7 2018. doi:10.24963/ijcai.2018/66. 65, 86
- [126] Monica Palmirani, Guido Governatori, Antonino Rotolo, Said Tabet, Harold Boley,

- and Adrian Paschke. Legalruleml: Xml-based rules and norms. In Frank Olken, Monica Palmirani, and Davide Sottara, editors, *Rule-Based Modeling and Computing on the Semantic Web*, pages 298–312, Berlin, Heidelberg, 2011. Springer Berlin Heidelberg. 15, 153
- [127] Sofia Panagiotidi, Sergio Alvarez-Napagao, and Javier Vázquez-Salceda. Towards the norm-aware agent: Bridging the gap between deontic specifications and practical mechanisms for norm monitoring and norm-aware planning. In *Revised Selected Papers of the COIN 2013*, volume 8386, pages 346–363. Springer-Verlag Inc., 2014. URL: http://dx.doi.org/10.1007/978-3-319-07314-9_19, doi:10.1007/978-3-319-07314-9_19. 39, 80, 119
- [128] Harshvardhan J. Pandit, Beatriz Esteves, Georg P. Krog, Paul Ryan, Delaram Golpayegani, and Julian Flake. Data privacy vocabulary (dpv) – version 2, 2024. arXiv:2404.13426. 30
- [129] Sharon Paradesi, Ilaria Llicardi, Lalana Kagal, and Joseph Pato. A semantic framework for content-based access controls. In *2013 International Conference on Social Computing*, pages 624–629, 2013. doi:10.1109/SocialCom.2013.94. 33
- [130] Archana Patel and Sarika Jain. Present and future of semantic web technologies: a research statement. *International Journal of Computers and Applications*, 43(5):413–422, 2021. arXiv:<https://doi.org/10.1080/1206212X.2019.1570666>, doi:10.1080/1206212X.2019.1570666. 19
- [131] Silvio Peroni and Fabio Vitali. Interfacing fast-fashion design industries with semantic web technologies: The case of imperial fashion. *Journal of Web Semantics*, 44:37–53, 2017. Industry and In-use Applications of Semantic Technologies. URL: <https://www.sciencedirect.com/science/article/pii/S1570826817300288>, doi:10.1016/j.websem.2017.06.001. 137, 138
- [132] Alessandro Ricci, Mirko Viroli, and Andrea Omicini. Give agents their artifacts: The a&a approach for engineering working environments in mas. In *Proceedings of the 6th International Joint Conference on Autonomous Agents and Multiagent Systems*, AAMAS '07, New York, NY, USA, 2007. Association for Computing

References

- Machinery. doi:10.1145/1329125.1329308. 16
- [133] E. Rissanen. extensible access control markup language (xacml) version 3.0, 2013. (accessed 15/02/2017). URL: <http://docs.oasis-open.org/xacml/3.0/xacml-3.0-core-spec-os-en.html>. 28
- [134] Victor Rodriguez-Doncel and Jaime Delgado. A media value chain ontology for mpeg-21. *IEEE MultiMedia*, 16(4):44–51, 2009. doi:10.1109/MMUL.2009.78. 26
- [135] Víctor Rodríguez-Doncel, Serena Villata, and Asunción Gómez-Pérez. A dataset of RDF licenses. In Hoekstra [80], pages 187–188. doi:10.3233/978-1-61499-468-8-187. 26
- [136] Víctor Rodríguez-Doncel, Joaquín Delgado, Sergio Llorente, Elena Rodríguez, and Luis Boch. Overview of the MPEG-21 media contract ontology. *Semantic Web Journal*, 7:311–332, 2016. URL: <https://doi.org/10.3233/SW-150184>. 26
- [137] Soheil Roshankish and Nicoletta Fornara. A methodology for formalizing different types of norms. In Dorothea Baumeister and Jörg Rothe, editors, *Multi-Agent Systems - 19th European Conference, EUMAS 2022, Düsseldorf, Germany, September 14-16, 2022, Proceedings*, volume 13442 of *Lecture Notes in Computer Science*, pages 348–363, Cham, 2022. Springer. doi:10.1007/978-3-031-20614-6_20. 89
- [138] K. Rowshankish and A. Trittipo. Data privacy: what every manager needs to know, 7 2018. visited on 2021-02-12. URL: <https://www.mckinsey.com/capabilities/risk-and-resilience/our-insights/data-privacy-what-every-manager-needs-to-know#/>. 131
- [139] Owen Sacco and Alexandre Passant. A privacy preference ontology (ppo) for linked data. In *LDOW*, 2011. URL: <https://api.semanticscholar.org/CorpusID:17336338>. 33
- [140] Pierangela Samarati and Sabrina Capitani de Vimercati. Access control: Policies, models, and mechanisms. In Riccardo Focardi and Roberto Gorrieri, editors, *Found-*

- dations of Security Analysis and Design*, pages 137–196, Berlin, Heidelberg, 2001. Springer Berlin Heidelberg. 12
- [141] Ravi Sandhu and Jaehong Park. Usage control: A vision for next generation access control. In Vladimir Gorodetsky, Leonard Popyack, and Victor Skormin, editors, *Computer Network Security*, pages 17–31, Berlin, Heidelberg, 2003. Springer Berlin Heidelberg. 13
- [142] Bastin Tony Roy Savarimuthu, Stephen Cranefield, Maryam Purvis, and Martin K. Purvis. Identifying conditional norms in multi-agent societies. In Marina De Vos, Nicoletta Fornara, Jeremy V. Pitt, and George A. Vouros, editors, *Coordination, Organizations, Institutions, and Norms in Agent Systems VI - COIN 2010 International Workshops, COIN@AAMAS 2010, Toronto, Canada, May 2010, COIN@MALLOWS 2010, Lyon, France, August 2010, Revised Selected Papers*, volume 6541 of *Lecture Notes in Computer Science*, pages 285–302. Springer, 2010. doi:10.1007/978-3-642-21268-0_16. 76, 96, 97, 114
- [143] John R. Searle. *The Construction of Social Reality*. Free Press, 1995. 16
- [144] Yoones A. Sekhavat. Privacy preserving cloth try-on using mobile augmented reality. *IEEE Transactions on Multimedia*, 19(5):1041–1049, 2017. doi:10.1109/TMM.2016.2639380. 131
- [145] Murat Sensoy, Timothy J. Norman, Wamberto Weber Vasconcelos, and Katia P. Sycara. OWL-POLAR: A framework for semantic policy representation and reasoning. *J. Web Sem.*, 12:148–160, 2012. doi:10.1016/j.websem.2011.11.005. 22, 31, 42, 59, 60, 65, 67, 69, 75, 86, 87, 92, 96, 103, 113
- [146] Marc Serramia, Maite Lopez-Sanchez, Juan A. Rodriguez-Aguilar, Javier Morales, Michael Wooldridge, and Carlos Ansotegui. Exploiting moral values to choose the right norms. In *Proceedings of the 2018 AAAI/ACM Conference on AI, Ethics, and Society*, AIES ’18, page 264–270, New York, NY, USA, 2018. Association for Computing Machinery. doi:10.1145/3278721.3278735. 93
- [147] Marc Serramia, Maite López-Sánchez, and Juan Antonio Rodríguez. A qualitative

References

- approach to composing value-aligned norm systems autores. 2020. doi:10.5555/3398761.3398904. 93
- [148] Robin Snyder. A practical introduction to the xml, extensible markup language, by way of some useful examples. visited on 2021-02-12. URL: <http://www.RobinSnyder.com>. 135, 140
- [149] Simon Steyskal and Axel Polleres. Towards Formal Semantics for ODRL Policies. In Bassiliades et al. [19], pages 360–375. doi:10.1007/978-3-319-21542-6_23. 27, 67, 90, 92
- [150] Alex Stolz. Clothing product information ontology language reference. visited on 2021-02-12. URL: <http://www.ebusiness-unibw.org/ontologies/cpi/ns>. 134
- [151] Hubert Tardieu. *Role of Gaia-X in the European Data Space Ecosystem*, pages 41–59. Springer International Publishing, Cham, 2022. doi:10.1007/978-3-030-93975-5_4. 13
- [152] Phillip J. Turner and Nicholas R. Jennings. Improving the scalability of multi-agent systems. In Tom Wagner and Omer F. Rana, editors, *Infrastructure for Agents, Multi-Agent Systems, and Scalable Multi-Agent Systems*, pages 246–262, Berlin, Heidelberg, 2001. Springer Berlin Heidelberg. 24
- [153] Philip Twu, Yasamin Mostofi, and Magnus Egerstedt. A measure of heterogeneity in multi-agent systems. In *2014 American Control Conference*, pages 3972–3977, 2014. doi:10.1109/ACC.2014.6858632. 24
- [154] Shambhu Upadhyaya. *Mandatory Access Control*, pages 756–758. Springer US, Boston, MA, 2011. doi:10.1007/978-1-4419-5906-5_784. 12
- [155] A. Uszok, J. M. Bradshaw, J. Lott, M. Breedy, L. Bunch, P. Feltovich, M. Johnson, and H. Jung. New Developments in Ontology-Based Policy Management: Increasing the Practicality and Comprehensiveness of KAoS. In *2008 IEEE Workshop on Policies for Distributed Systems and Networks*, pages 145–152, 2008. doi:10.1109/POLICY.2008.47.

References

- [156] Andrzej Uszok, Jeffrey Bradshaw, R. Jeffers, Niranjani Suri, Patrick Hayes, Maggie Breedy, Larry Bunch, Matthew Johnson, Saeed Kulkarni, and James Lott. Kaos policy and domain services: toward a description-logic approach to policy representation, deconfliction, and enforcement. pages 93–96, 07 2003. doi:10.1109/POLICY.2003.1206963. 11, 91
- [157] Andrzej Uszok, Jeffrey M. Bradshaw, James Lott, Maggie R. Breedy, Larry Bunch, Paul J. Feltoch, Matthew Johnson, and Hyuckchul Jung. New Developments in Ontology-Based Policy Management: Increasing the Practicality and Comprehensiveness of KAOs. In *POLICY 2008, 2-4 June 2008, Palisades, New York, USA*, pages 145–152, 2008. doi:10.1109/POLICY.2008.47. 32, 59, 61
- [158] Francesco Viganò, Nicoletta Fornara, and Marco Colombetti. An event driven approach to norms in artificial institutions. In Olivier Boissier et al., editors, *Coordination, Organizations, Institutions, and Norms in Multi-Agent Systems. AAMAS 2005*, volume 3913 of *LNCS*, pages 142–154. Springer Berlin Heidelberg, 2006. 61
- [159] Guest Editor: Serena Villata, editor. *Special Issue: Normative Multi-Agent Systems*, volume 5 of *Journal of Applied Logics - IfCoLog Journal*. College Publications, 2018. 39
- [160] Serena Villata, Michal Araszkiewicz, Kevin D. Ashley, Trevor J. M. Bench-Capon, Luther Karl Branting, Jack G. Conrad, and Adam Wyner. Thirty years of artificial intelligence and law: the third decade. *Artif. Intell. Law*, 30(4):561–591, 2022. URL: <https://doi.org/10.1007/s10506-022-09327-6>, doi:10.1007/S10506-022-09327-6. 127
- [161] G. H. von Wright. Deontic logic. *Mind, New Series*, 60(237):1–15, 1951. 59, 77, 115
- [162] Georg Henrik von Wright. *Norm and Action: A Logical Enquiry*. Routledge and Kegan Paul, 1963.
- [163] Michael Wooldridge. *An Introduction to MultiAgent Systems*. Wiley Publishing, 2nd edition, 2009. 1

- [164] Jing Xie and Chen-Ching Liu. Multi-agent systems and their applications. *Journal of International Council on Electrical Engineering*, 7(1):188–197, 2017. arXiv:<https://doi.org/10.1080/22348972.2017.1348890>, doi: 10.1080/22348972.2017.1348890. 16
- [165] N.-P. Yu and C.-C. Liu. Multiagent systems. In *Advanced Solutions in Power Systems: HVDC, FACTS, and Artificial Intelligence*, pages 903–930. John Wiley & Sons, Hoboken, NJ, 2016. 16
- [166] Franco Zambonelli, Nicholas R. Jennings, and Michael Wooldridge. Developing multiagent systems: The gaia methodology. *ACM Trans. Softw. Eng. Methodol.*, 12(3):317–370, jul 2003. doi:10.1145/958961.958963. 16
- [167] Alexandra-Madalina Zarafin, Antoine Zimmermann, and Olivier Boissier. Integrating semantic web technologies and multi-agent systems: A semantic description of multi-agent organizations. In *International Conference on Agreement Technologies*, 2012. URL: <https://api.semanticscholar.org/CorpusID:34839899>. 21
- [168] Mirko Zichichi and Víctor Rodríguez-Doncel. Encoding of media value chain processes through blockchains and mpeg-21 smart contracts for media. *IEEE Multi-Media*, 30(4):26–35, 2023. doi:10.1109/MMUL.2023.3303393. 26