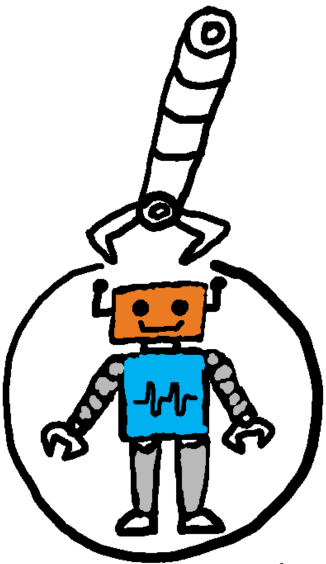
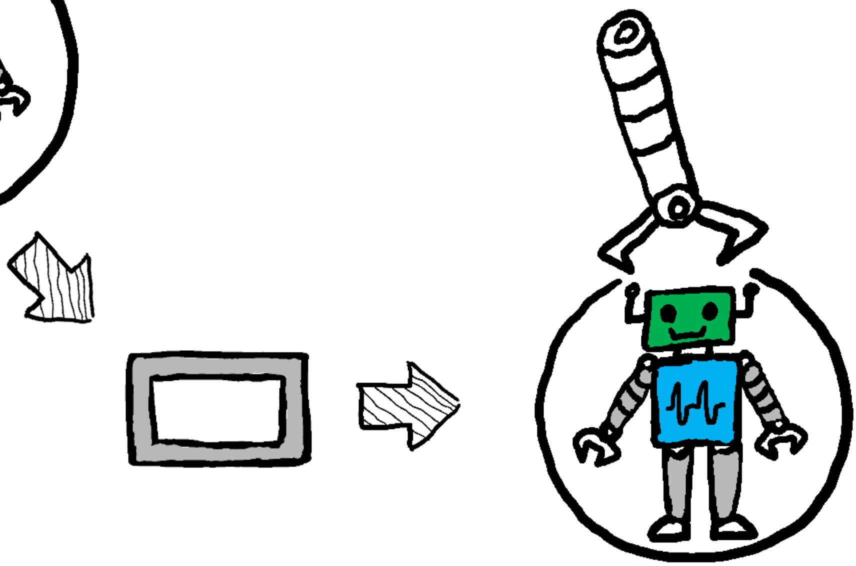
ORGANIZATIONAL COMPLIANCE



An agent-based model for designing and evaluating organizational interactions



JIE JIANG

Organizational Compliance

An agent-based model for designing and evaluating organizational interactions

Jie Jiang

Organizational Compliance: An agent-based model for designing and evaluating organizational interactions

Proefschrift

ter verkrijging van de graad van doctor
aan de Technische Universiteit Delft,
op gezag van de Rector Magnificus prof. ir. K.C.A.M. Luyben,
voorzitter van het College voor Promoties,
in het openbaar te verdedigen op vrijdag 20 februari om 12:30 uur

door

Jie JIANG

Master of Engineering, Xi'an Jiaotong University
geboren te Shaanxi, China.

Dit proefschrift is goedgekeurd door de promotor:

Prof. dr. Y.H. Tan

Copromotoren:

Dr. M.V. Dignum

Dr. H.M. Aldewereld

Samenstelling promotiecommissie:

Rector Magnificus,	voorzitter
Prof. dr. Y.H. Tan,	Technische Universiteit Delft, promotor
Dr. M.V. Dignum,	Technische Universiteit Delft, copromotor
Dr. H.M. Aldewereld,	Technische Universiteit Delft, copromotor
Prof. dr. C.M. Jonker,	Technische Universiteit Delft
Prof. dr. ir. W.M.P. van der Aalst,	Technische Universiteit Eindhoven
Prof. dr. L. van der Torre,	University of Luxembourg
Prof. dr. M.P. Singh,	North Carolina State University
Prof. dr. ir. M.F.W.H.A. Janssen,	Technische Universiteit Delft, reservelid

SIKS Dissertation Series No. 2015-08

The research reported in this thesis has been carried out under the auspices of SIKS, the Dutch Research School for Information and Knowledge Systems.

ISBN 978-94-6186-431-4

Printed by: CPI Koninklijke Wöhrmann
Cover design by: Jie Jiang and Wen Li

Printed in the Netherlands.

Acknowledgements

This dissertation is a result of exploration, frustration, communication, encouragement and persistence, which makes a unique chapter of my life. I want to acknowledge all the people that have gotten me here.

First, I would like to thank my promotor Prof. Yao-hua Tan who provided me the opportunity to study a PhD in TU Delft and introduced me to the most excellent supervisor Dr. Virginia Dignum. I could not express my gratitude to Virginia by words since she is not only my supervisor in study but also my mentor in life. Believe it or not, she always asked me to stop working and take holidays. Moreover, she introduced me to my other excellent supervisor Dr. Huib Aldewereld who always held patience in our long discussions and helped me a lot with scientific as well as technical problems. Over the past four years, I really learned a lot from them in both methodologies and technologies, which will be a treasure for my future career and life.

Special thanks to Prof. Munindar Singh, Prof. Leon van der Torre, Prof. Wil van der Aalst, Prof. Catholijn Jonker and Prof. Marijn Janssen. Thank you all very much for joining in my doctoral committee and the insightful reviews of the dissertation. I also want to thank Bob Huisman, Frank Dignum, Julian Padget, Marina De Vos, John Thangarajah, Sietse Overbeek, and Ziv Baida who have worked with me on various papers. Besides, I would like to thank the people from the peer group of AAMAS goers (Diederik, Reyhan, Amineh, Tina, Gleb . . .)

My colleagues in ICT section are all wonderful people. I am very happy to have shared an office with Bram, Janneke, Thomas, Reza, Sheba, Mohsen, and Klara. It was a lot of fun learning about each other's research and discussing different cultures. In particular, I would like to thank Janneke who was my officemate for the longest time and taught me a lot of things about Dutch culture. From "next doors", I also got a lot of support from Anneke, Wally, Fatemeh, Jolien, Joris, Ebrahim, Devender, and Potchara. The secretaries Jo-Ann, Eveline and Laura helped me a lot with practical matters. Thank you all!

It would be a bumpy journey in the dark without the love and support from family and friends. I am really grateful to my parents who provides me with anything I need to chase after my dream. Special thanks to my brother and his wife who have been accompanying my parents while I am thousands of kilometers away. I also want to thank my dearest friend Tingting who has given me strength and encouragement along the way. There are many more friends than I could possibly mention here. I thank you all for being there with me. Last but not the least, I would like to thank my wonderful husband Wen who is the best companion both in study and life. I will never forget our promise: “Whenever you look back, I will be there for you.”

谨以此书答谢父母的养育教导之恩！

Jie Jiang（姜 洁）

Delft, January 2015

Contents

Chapter 1

Introduction

Globalization is leading to a dramatic increase in both the amount and complexity of rules that direct and regulate the behavior of organizations, such as guidelines, specifications and legislation, etc. Organizations are under the pressure to take measures to ensure that their business operations are complying with these rules. Why is such compliance important for organizations? The answers can be well reflected from the consequences of non-compliance, which are not limited to statutory or legal penalties [211, 31]. The indirect costs of non compliance to an organization are often more significant, including damage to the organization's reputation or credit rating, loss of contracts, the inconvenience and cost of righting a mistake, and even being not allowed to operate any longer [99, 24]. Moreover, compliance to, for example, business agreements and governmental regulations, is essential to ensure the success of participants' interactions within as well as across organizations, the safety of business environments and the social welfare [116]. In the context of international trade, compliance to, for example, control requirements and information transparency, enables business organizations to show to the government that they are in control of their business operations, which makes them well positioned to obtain a trusted trader status and the related benefits of trade facilitation [201]. As such, it is essential for organizations (both public and private) to take measures to achieve and ensure their compliance.

Organizations are prevalent in every arena of human life, and they are engaged in performing highly diverse tasks. As Parsons [167] puts it, "the development of organizations is the principal mechanism by which, in a highly differentiated society, it is possible to achieve goals beyond the reach of the individual." This is implied by the fact that organizations provide the setting for a wide variety of social processes, including socialization, communication, the formation of norms, the exercise of power, and goal setting and attainment. Moreover, organizations are characterized by distinctive structural arrangements that affect the operation of the processes occurring within them [109, 186]. Thus, organizations can be seen as distinctive structural arrangements that guide and regulate the collaborative pursuit of some specified objectives by means of various social processes.

Organizational regulation is rarely self-sufficient to ensure the achievement of organizational objectives, especially in a multi-organizational interaction setting, but depends on various (external) regulation sources such as legal regulations, business agreements, industrial best practices, etc. For example, when dairy products are exported, apart from the internal process control of the dairy exporter, many other sources of regulations are imposed by agencies such as customs which regulates the activities concerning export declaration, transportation and Value-Added Tax (VAT) settlement, health agency which mainly regulates the activity of health certification, agriculture agency which puts information requirements on export declaration, and tax agency which regulates the activities of VAT settlement and invoicing [102]. Given the diversity of information sources and possibly conflicting interests, it is likely that organizational regulation is not in accordance with other regulation sources. Moreover, organizational participants (human, software and organizations[1]) are autonomous entities, as they have the capability to act independently and control over their own actions. It is possible that the behavior of the participants deviates from what is desired and prescribed by the regulations relevant to the operation of the organization [205, 150]. For example, fraud with VAT and excise in the European Union amounts to tens of billion euros per year [201]. This possibility raises an important organizational issue: "compliance". Compliance, in general, may be understood as either a state of being in accordance with established guidelines, specifications and legislation, or the process of becoming so [193]. From an organizational perspective, compliance may refer to

- the consistency between the compliance requirements of the organizational regulation and that of other regulation sources relevant to the operation of the organization, and
- the consistency between the organizational participants' behavior and the compliance requirements of all the regulation sources.

As enterprises increasingly rely on business process models and execution environments to manage and automate their business operations, it is a common way to employ approaches to ensure compliance in the design and execution of business processes [128, 63]. Current approaches focus more on the choice and ordering of business operations but provide little knowledge about how to handle the dynamics of organizational compliance with respect to changing environments/contexts, how to deal with the combination of multiple sources of compliance requirements with respect to consistency, and how to facilitate the reuse of solutions cross different contexts.

[1] According to Coleman [44], organizations must also be viewed as actors in their own right. They can take actions, utilize resources, enter into contracts, and own property.

Organizational participants are those individual actors who, in return for a variety of inducements, make contributions to the accomplishment of an organization's objectives [18]. They are expected to take part in the organization and follow the organizational arrangements such that the objectives of the organization can be fulfilled. Without participants/actors, organizations do not have a real existence. However, as we mentioned earlier, participants are autonomous and have their personal interests and needs. It is possible that the behavior of the participants deviates from the compliance requirements of the organization. In this sense, individual autonomy is also a key factor for the achievement of organizational compliance. An essential question therefore is how to find the right balance between the level of control of the organization and the level of autonomy of the participants.

To these ends, organizations need the means to ensure their compliance through the specification of coordination and control components, structures, processes, and actors (known as organizational design) such that

- dynamics, diversity, and reusability issues of organizational compliance can be managed, and
- organizational control and individual autonomy can be balanced in achieving organizational compliance.

Formalisms are a suitable means to provide a precise understanding of the formation of organizational compliance, and enable an accurate evaluation of organizational designs. In this research, we are concerned with organizational compliance in the setting of virtual organizations [163] where autonomous participants from multiple organizational entities interact with each other to achieve the organizational objectives. We use Multi-Agent System (MAS) theories as a formalization of such a problem because of its capability in dealing with autonomous behavior of individual participants as well as the coordination of the participants' interactions [216]. Colored Petri Nets (CPNs) are used as a verification mechanism to facilitate the evaluation of organizational compliance because of its capability in modeling and validating concurrent systems and the well-developed techniques and tool support [112].

This chapter is organized as follows. In Section 1.1, we present the research questions of this research. The research approach used for this research is illustrated in Section 1.2. Finally, we conclude this chapter by presenting the outline of the following chapters in Section 1.3.

1.1 Research Question

As illustrated above, compliance is an important factor for organizations to maintain their business competitiveness, ensure the success of the participants' interactions within the organization as well as with other organizations, and guarantee the safety of the whole business environment and social welfare. Organizations need the means to ensure their compliance through organizational design such that dynamics, diversity, and reusability issues of organizational compliance can be managed, and organizational control and individual autonomy can be balanced in achieving organizational compliance, which leads to the main research question:

- How to ensure organizational compliance through organizational design, taking into account various regulation sources and the environmental characteristics, and the autonomy of organizational participants?

To answer this question, the first important task is to identify the constructs and structures that are needed to represent organizational compliance through organizational design, which leads to the first research question.

***Research Question 1*:** What are the constructs and structures that are needed for the representation of organizational compliance in organizational design?

Organizations are complex systems with a fundamental feature of hierarchy in the form of clustering and levels [186]. This is reflected by the fact that organizations are usually made up of roles contained within work groups, within departments, within divisions, which in turn may be seen as roles within larger organizations [44]. Such organizational structures reflect important features borrowed from or impressed on them by the context (a specific physical, technological, cultural, and social environment) to which the organization must adapt. This implies that to coordinate and regulate the behavior of an organization, it may be as important to look outside the organization and its context as to look inside the organization at its component units. Accordingly, the establishment of organizational compliance needs to cover multiple levels of the participants' activities and interactions through their enacting roles, and reflect upon the specific contexts where the activities and interactions occur. In this way, the compliance responsibilities of organizational participants can be encapsulated into modular and meaningful blocks that reflect the changes of requirements of organizational compliance with respect to the changing contexts. Moreover, the modularization of coordination and control components facilitates the reuse of solutions for ensuring organizational compliance across different contexts. Chapter 3 and 4 provide an answer to the first research question.

Compliance refers to the regularized aspects of organizational design which provide an accurate description of the "right" ways of operation. From an organizational perspective, it is expected that the participants follow the compliance requirements that are relevant to their enacting roles in the organization such that the objectives of the organization can be achieved in a predictable way [119]. However, since participants are autonomous entities and organizations do not have a full control over their actions, it is necessary to evaluate whether the actual behavior of the participants is in accordance with what is desired and prescribed by the organization, which leads to the second research question.

***Research Question 2*:** How can we evaluate the actual behavior of organizational participants with respect to the compliance requirements of organizational regulation?

Participants' compliance implies their actual behavior being in accordance with all sorts of rules governing their behavior that specify, in particular, appropriate means for pursuing the objectives of the organization. These rules are applied to the organizational participants through the roles they enact in the organization, and may come from various sources such as legal regulations, business contracts, internal policies, and industrial best practices. They can be fairly complex in terms of the conditions, targets and scopes they refer to. When taking a system point of view, the possibility of interrelationships between them brings added complexity to the understanding of organizational compliance [116]. Moreover, as Searle [188] argued, there is a distinction between what he called "regulative" and "constitutive" rules. Regulative rules concern what ought to be the case by regulating antecedently existing activities. Constitutive rules do not merely regulate but also create the very possibility of certain activities. To these ends, the evaluation of participants' compliance should consider all these aspects of rules that govern the behavior of the organizational participants. Chapter 5 provides an answer to the second research question.

Besides the compliance of organizational participants, another important aspect of organizational compliance is the consistency among different sources of compliance requirements that are relevant to the achievement of organization compliance in a larger context. For example, in virtual organizations, multiple institutions[2] may be employed to cover different aspects of regulating the behavior of the participants in order to achieve the virtual organizations' goals [140]. With possibly overlapped governance on the behavior of organizational participants, these different institutions may have contradictory regulations. Therefore, it is necessary to evaluate whether there are any conflicts between different sources of compliance requirements, which leads to the third research question.

***Research Question 3*:** How can we evaluate the consistency of different sources of compliance requirements that are relevant to the regulation of organizational behavior?

In practice, every source of compliance requirements for the regulation of organizational behavior has its boundary of applicability. For example, some international trade regulations are only effective for certain goods (e.g., AEO[3] certified goods). That is to say, with respect to a particular source of compliance requirements, some combinations of behavior and metadata are meaningful and others

[2]We follow the definition of Ostrom [165] that institutions are the prescriptions that humans use to organize all forms of repetitive and structured interactions.

[3]The Authorized Economic Operator (AEO) is a European-wide customs initiative that aims to secure the supply chain while at the same time reducing the administrative burden for actors through the use of self-regulation.

are not. Therefore, in (virtual) organizations governed by multiple institutions, the evaluation of the consistency of different sources of compliance requirements needs to take into account the applicable boundary of the originating institutions. Moreover, as stated before, compliance requirements may have various interrelations, e.g., sanction, reparation. Consistency evaluation thus also needs to consider how the interrelations between compliance requirements impact the evaluation results. Chapter 6 presents an answer to the third research question.

Organizational compliance is the "ideal" status that organizations expect to achieve and maintain, and the achievement of organizational compliance largely depends on the participants who finally take actions. However, organizational participants are autonomous entities with various values and objectives. The autonomy nature of the participants implies that they have the tendency to act upon individual interests and needs, which is important to ensure personal satisfaction and involvement [76, 101, 41]. To this end, the autonomy of the participants should not be ignored in achieving organizational compliance, and organizational design needs to seek for a balance between the stability of organizational control and the flexibility of individual operation, which leads to the fourth research question.

***Research Question 4*:** How to take into account the autonomy of individual participants in the evaluation of organizational compliance?

The autonomy of individual participants indicates their personal preferences, and influence their choices of actions to fulfill their roles in the organization. These preferences could be conceived of as an individual's attitude towards a set of alternatives [142]. In this sense, preferences can be an effective way to represent the autonomy desired by individual participants in organizational interactions. Therefore, to be able to incorporate individual autonomy in the evaluation of an organizational design, a mechanism of formalizing and evaluating the preferences of organizational participants is needed. Chapter 7 gives an answer to the fourth research question.

Given the four research questions elicited above, we now illustrate the research approach adopted in this dissertation.

1.2 Research Approach

In this research, we aim at developing a framework named OperA+ for formalizing and evaluating organizational compliance in the setting of virtual organizations where autonomous participants interact with each other to achieve organizational objectives. Following the design science framework proposed in [104], this research consists of three research cycles, as shown in Figure 1.1.

The **relevance cycle** links the research to concrete application domains that not only provide specific requirements but also criteria for the evaluation of the research results. In this research, we mainly focus the application domains on the

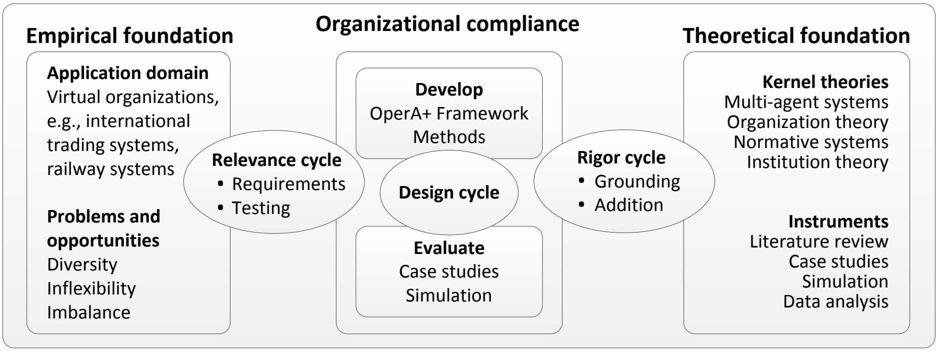


Figure 1.1: *Research Approach.*

setting of virtual organizations (such as international trade and railway operation) in which a number of individual organizations work together to achieve their own objectives as well as some collective goals. The evaluation criteria are flexibility and reusability of organizational design, in which flexibility concerns the capability of ensuring organizational compliance in changing environments (contexts), reusability concerns the capability of reusing solutions of ensuring organizational compliance across different contexts.

The **rigor cycle** provides a knowledge base for the research and connects theoretical foundations with the research activities, including the analysis of the state-of-the-art research and ensure that the results lead to scientific contributions. In our research, a main observation is that organizations and their interactions with each other towards the organizational objectives can be modeled as a multi-agent system where intelligent agents interact together within a structured environment. Therefore, MAS related theories and methodologies provide potential supports for this research, such as agent organization modeling, normative systems and agent reasoning. In addition, the specific characteristics of this research requires advanced methodologies such as organization theories, institution theories, concurrent system modeling and verification, which in turn will contribute to MAS.

The central **design cycle** iterates between the core activities of developing and evaluating the design artifacts and the processes of the research. The output must also be returned into the empirical foundation for refining the design until it well fits the research objective and the specific requirements of the application domains. The design artifacts in this research are the OperA+ framework with a suite of formalisms and methods for formalizing and evaluating organizational compliance. In order to assess the artifacts, we have conducted several case studies together with domain experts to analyze its efficiency and applicability.

1.3 Overview of the Thesis

This dissertation mainly consists of four parts. The first part, consisting of Chapter 1 and 2, presents the motivation and theoretical background of this research. The second part, consisting of Chapters 3 to 7, presents the proposed OperA+ framework and its constituting structures by illustrating the design choices and the features of the components as well as their formalizations and operationalizations. The third part, consisting of Chapters 8, 9 and 10, presents the application studies of OperA+. Finally, the fourth part, Chapter 11, summarizes the research results and identifies the directions for future work. The remainder of the dissertation is structured as follows.

Chapter 2 presents the kernel theories related to this research by a discussion of the concepts that are relevant for the design and implementation of organizational models to ensure organizational compliance.

Chapter 3 gives an overview of the proposed framework OperA+ by illustrating its outline and important features.

Chapter 4 illustrates the social structure of OperA+, which specifies how social entities and their relations are captured and formalized in order to provide a way of encapsulating compliance requirements.

Chapter 5 illustrates the normative structure of OperA+, which presents how compliance requirements are captured and formalized to enable the compliance checking of organizational participants' behavior.

Chapter 6, based on the description in Chapter 5, investigates the combination of multiple sources of compliance requirements and presents how conflicts between the compliance requirements can be detected.

Chapter 7 illustrates the preference structure of OperA+, which specifies how preferences of organizational participants are formalized, and how the satisfaction of individual preferences and the fulfillment of compliance requirements are combined in the evaluation of organizational behavior.

Chapter 8 presents two application cases to evaluate the social structure of OperA+.

Chapter 9 presents three application cases to evaluate the normative structure of OperA+, in which the first two cases focus on the evaluation of the compliance checking aspect of the normative structure and the third case focuses on the evaluation of the conflict detection aspect of the normative structure.

Chapter 10 presents an application case to evaluate the preference structure of OperA+, and the integration of individual preferences with normative constraints in the evaluation of organizational behavior.

Chapter 11 presents the conclusions and identifies directions for future work.

Chapter 2

Background

Research is never carried out behind closed doors but always learns from the success and failure of previous work. This is also the case of the research in this dissertation. Before formally presenting our proposal, this chapter gives an overview of the related theories and research areas upon which this research is built. It will, on the one hand, help to gain a better understanding of the concepts and ideas described in the following chapters, and on the other hand, help to reveal the opportunities for this research.

This chapter is organized as follows. In Section 2.1, we describe the major perspectives that have been adopted by researchers in organizational analysis and provide the stance of this research. Section 2.2 gives an analysis of the definition of compliance and the approaches for ensuring compliance in the literature. In Section 2.3 the main aspects of the agent paradigm are discussed. In Section 2.4, we illustrate the concept of roles and discuss the work on modeling roles. In Section 2.5, we illustrate the concept of norms and the main aspects in the study of norms. Thereafter, work on normative multi-agent systems is presented in Section 2.6. Section 2.7 looks at the existing approaches for norm compliance checking and conflict detection. Section 2.8 introduces the model of Colored Petri Nets. In Section 2.9, work on the combination of Petri Nets and multi-agent systems is discussed. Finally, conclusions are presented in Section 2.10.

2.1 Organizations

Organizations were present in ancient civilization but flourished in modern societies. They are engaged in performing highly diverse tasks such as public administration, production and distribution of goods, provision of services, preservation of culture. As Parsons put it, "the development of organizations is the principal mechanism by which, in a highly differentiated society, it is possible to 'get things done', to achieve goals beyond the reach of the individual" [167].

The study of organizations is not only a specialized research field within the discipline of sociology but also a focus of multidisciplinary research. In sociology, the emergence of the field of organizations may start from the translation into English of Weber's work [212, 213]. Around the same time, Herbert Simon took the lead of an important interdisciplinary development on building a behaviorally oriented science of administration, and emphasis was then placed on decision making and choice within organizations [192]. According to the study in [186], there are three major perspectives in the analysis of organizations:

- rational system: organizations are collectivities oriented to the pursuit of relatively specific goals and exhibiting relatively highly formalized social structures.
- natural system: organizations are collectivities whose participants share a common interest in the survival of the system and who engage in collective activities, informally structured, to secure this end.
- open system: organizations are systems of interdependent activities linking shifting coalitions of participants; the systems are embedded in–dependent on continuing exchanges with and constituted by–the environments in which they operate.

These three perspectives provide contrasting paradigms for analyzing organizations. The rational system perspective stresses the importance of control, and the theorists with this perspective focus on the normative structure of organizations (the specificity of goals and formalization of rules and roles). The natural system perspective stresses the importance of the characteristics of participants, and the theorists with this perspective focus on behavior structure (activities, interactions, and sentiments). The open system perspective stresses the importance of organizations' interchange with their environment, and the theorists with this perspective focus on complexity and variability of both individual participants and subgroups as well as the looseness of connections among them.

In this research, we take a combined view for the analysis of organizations. Firstly, we regard organizations as purposeful and the cooperation among participants can be structured (to the extent that the *rules* governing behavior are precisely and explicitly formulated and that *roles* and role relations are prescribed independently of the personal attributes of individuals occupying positions in the structure). Secondly, we regard participants as autonomous entities who have differing interests and needs. Thirdly, we contend that organizations are not independent but operate in a specific environment to which they must adapt.

2.2 Compliance

The research on compliance has a long history, which can be dated back to 1600s when Thomas Hobbes touched on the compliance problem in contractarian ethics. He stated that compliance with contracts may be better for a whole group and it is in the individuals' best interest to agree to the contracts but it may not be in their interest to actually comply to them [106]. Such a stance also holds in an organizational setting, as compliance with organizational arrangements may be in the best interest of the organization as a whole but may not be in line with the actual needs of the individual participants in the organization. Therefore, it is necessary to explore effective measures to pursue and monitor compliance.

To be able to pursue and monitor compliance, first one needs to know what compliance is. In the literature, there has been a number of proposals for the definition of compliance. For example, in the study of how treaties induce compliance [156], compliance is defined as "an actor's behavior that conforms to a treaty's explicit rules". In the study of environmental compliance and sustainable development [222], compliance is defined as "a state of conformity or identity between an actor's behavior and a specified rule, regardless of the motivations, circumstances, or causes that lead to that conformity". In a specific context of Information Technology (IT) compliance of industrial information systems [129], compliance is defined as an accordance of corporate IT systems with predefined policies, procedures, standards, guidelines, specifications, or legislation. From an organizational perspective in general [68], compliance is considered as a state of accordance between an actor's behavior or products on the one side, and predefined explicit rules, procedures, conventions, standards, guidelines, principles, legislation or other norms on the other side. Focusing primarily on judicial decision making, Edelman and Talesh argued that the nature of organizational compliance is best illustrated not by a compliance versus noncompliance dichotomy, but rather by a processual model in which organizations construct the meaning of both compliance and law [57]. Finally, using a dictionary definition [193], compliance means the action or fact of complying with a wish or command. From all these definitions, two important aspects of compliance can be abstracted. First, the establishment of compliance is a process of identifying all sorts of rules that contribute to the meaning of compliance. Second, the evidence of compliance is a state of being in accordance with those rules. In this research, we investigate both aspects.

As for the solution of ensuring compliance, it is recognized that there is a distinction between two types of approaches in the literature [108, 214, 57]. The first type of approaches takes a *rationalist* view and focus on the actor's calculation of benefits and costs in determining whether or not to comply. Game theory is widely adopted in these approaches to investigate behavioral motivations. A major mechanism of ensuring compliance is enforcement in which undesired behavior is discouraged by means of punishment while desired behavior is stimulated by means of rewards. The second type of approaches takes a *normative* view and focuses on mechanisms of coordination and regulation to promote compliance. Concepts, such as roles, obligations, prohibitions, commitments etc., are usually employed in these approaches. Actors are expected to follow the rules that are relevant to their roles and the rules need to be internalized as legitimate by the actors. Non-compliance in this case may be due to the reason that the rules are complex, ambiguous or changing continuously, and the actors lack in capacity, knowledge or commitment.

In this research, we take a normative view on the study of compliance since our analysis is mainly from an organizational perspective without assuming the knowledge of individual actors' decision making process. However, we do not require that the rules need to be internalized by the actors but take into account their personality in the normative design.

2.3 Agents

According to the most widely accepted definition, an agent is "an encapsulated computer system that is situated in some environment and that is capable of flexible, autonomous action in that environment in order to meet its design objectives" [217]. According to this definition, several properties of agents are distinguished.

- reactive: the ability of maintaining an ongoing interaction with the environment and responding to changes that occur in it,
- pro-active: the ability of taking the initiative in performing actions to achieve goals,
- social: the ability of interacting with other agents including humans,
- autonomous: the ability of acting independently and controlling its own behavior and internal state.

The first three properties are an indication of the agents' ability of performing flexible actions. Moreover, the notion of environment implies that there is a clear boundary between the agent and its situated environment which the agent can sense and effect. Among these four properties, autonomous is the focus of this research, as it determines the very possibility that agents are able to choose to deviate from the compliance requirements.

For the implementation of agents, a number of architectures have been proposed in the literature. According to the classification presented in [215], there are in general four types of agent architectures: (1) logic-based agents in which decision making is realized by logical deduction [78, 139], (2) reactive agents in which decision making is realized by some direct mapping from situation to action [36, 148], (3) belief-desire-intention (BDI) agents in which decision making is based on the manipulation of data structures representing the beliefs, desires and intentions of the agent [33, 175], and (4) layered agents in which decision making is realized via multiple interacting layers [158, 67]. Among these architectures, the BDI architecture is well suited for the modeling of organizational participants, as it is intuitive and generic enough to represent both natural and artificial agents, and it provides a clear functional decomposition in building an agent.

The BDI architecture, inspired from the philosophical theories of Bratman [32], is based on the idea that an agent has certain mental attitudes of belief, desire and intention, respectively representing the information, motivational, and deliberative states of the agent. The reasoning of BDI agents involves two important processes: deciding what goals to achieve, and how to achieve these goals. In order to assist agents' decision making when alternative plans are available, the notion of preference is integrated in the BDI framework (e.g., [48], [210]). The desires of an agent are considered as a reflection of its long term preferences [35].

In this research, we do not incorporate a complete representation of agents as we mainly focus on the organizational aspects and assume that organizations do not have a full control of the participants. But in order to reflect the characteristics of individual participants, we provide a mechanism to integrate agent preferences in the analysis of organizational interactions.

2.4 Roles

The nature of roles and the way of modeling them have been studied in various domains such as sociology and philosophy, knowledge representation and engineering, conceptual modeling, multi-agent systems, etc. [151, 26].

In sociology, the study of roles focuses more on human actors. As argued by Biddle [20], roles are "those behaviors characteristic of one or more persons in a context". In this domain, many issues regarding roles have been studied such as role consensus, role playing and role conformity. Role consensus focuses on the agreement of expectations from different persons. The different expectations on the same role may introduce role conflict, which in normative systems may result in conflict verdict on a person's behavior. Role playing regards a person's imitation of the behavior expected for a role. It generally defines the relation between a person and the roles the person plays. Role conformity evaluates whether a behavior is similar to or determined by the role expectation. Normative systems focuses on the aspect of role conformity that concerns whether role enactors are complying with the role inscription. In an organizational context, roles are viewed as expectations for or evaluative standards in assessing the behavior of participants occupying specific social positions [186], and role behavior refers to the recurring patterns of actions in individuals' interrelated activities so as to yield a predicable outcome [126]. The concept of roles in this context is closely related to norms, goals/objectives, activities, etc.

The way of modeling roles has a long history in object-oriented and conceptual modeling. An early work is the role data model proposed by Bachman and Daya [17] which explicitly introduced the notion of roles. It is based on the observation that most conventional file records are role oriented, e.g., these files typically deal with employees, customers, patients, or students, all of which are role types. In [198], Sowa proposed a distinction between *natural types* that relate to the essence of the entities and *roles types* that depend on accidental relationship between entities. In his subsequent work, Sowa asserted that role types are subtypes of natural types [199]. For example, the role type Child is a subtype of the natural type Person. An ontological distinction between role and natural types was further presented by Guarino: an individual of a role type has to stand in relation to other individuals while an individual of a natural type does not [96]. In the well-known Universal Modeling Language, roles are represented as labels of the entity types linked by a specific relation [71]. This line of research focuses on issues such as dynamic and multiple classification of objects, object collaboration, polymorphism, and substitutability, etc. A review of the most contemporary literature can be found in [200].

In multi-agent systems (MAS), roles are generally viewed as expectations of the agents' actions and interactions. The concept of roles under this context is closely linked to concepts such as organizations, obligations, tasks, goals, plans, etc. For example, in a model for the specification of organized collective agency [166], roles are seen as corresponding to qualities that agents might have. They are used as a high-level mechanism for structuring the desired behaviors by associating roles with deontic notions that describe the obligations and permissions of the agents that can play such roles. In the Gaia methodology for developing multi-agent systems [223], roles are defined in terms of functionalities, activities, and responsibilities, as well as interaction protocols and patterns. Taking an organizational perspective on the analysis of MAS, both approaches point out that an organization is independent of the agents enacting its roles. Such a perspective on roles can be widely found in the literature of organization-oriented models for developing multi-agent systems, which will be discussed in Section 2.6.1.

In this research, we take the perspective of role representation in MAS as a basis for organizational design, given the close link between how roles are formalized in MAS and how roles are perceived in organizations. In particular, we focus more on the aspects of roles that are essential for facilitating organizational compliance/conformity.

2.5 Norms

The subject of norms has been studied in many different research fields ranging from social philosophy, psychology and sociology, to legal theory and computer science. For example, according to philosophy, a norm is an authoritative rule or standard by which something is judged and, on that basis, approved or disapproved (Columbia Encyclopedia). In sociology, a norm is seen as a rule or standard of behavior shared by members of a social group (Encyclopedia Britannica). In computer science, especially agent community, a norm is a prescription of how the agents ought to behave, and specify how they are permitted to behave and what their rights are [123]. Moreover, according to many studies in legal and social theory, a distinction is recognized between regulative and constitutive norms (rules) [177, 100, 6, 188]. Regulative norms concern what ought to be the case (e.g., "vehicles are forbidden to enter public parks"). Constitutive norms concern what counts as what in a given context (e.g., " horses count as vehicles in public parks").

2.5.1 Representation of Norms

Norms in practice are usually expressed in natural languages (but in jargon). In order to reason about normative consequences, there is a need to capture the precise meaning of norms in a formal representation.

As for regulative norms, deontic logic is a widely used approach, which studies logical relations among obligations, permissions, and prohibitions [154]. The first formalism of this kind is the standard deontic logic (SDL) based on the work of von Wright [218]. An interesting extension is the Dyadic Deontic Logic [219], which introduces temporal relations in deontic expressions. Other extensions including [189] and [153] which propose to apply deontic modalities exclusively over actions instead of applying deontic modalities over formulas (or state of affairs) as in the case of SDL; [54] which proposes a temporal deontic logic that facilitates reasoning about obligations and deadlines. Borrowed the idea of representing deontic aspects by preferences from deontic logic [202], Raskin et al. proposed to represent ideal and sub-ideal deontic behavior in Petri nets by extending these nets with a preference relation [176]. On the basis of these primitive logical forms, a number of more advanced logical languages have been proposed in the literature. For example, Governatori et al. proposed a contract specification language [84, 86], obtained from the combination of three logical components: Defeasible Logic, deontic concepts, and a fragment of a logic to deal with normative violations. $\mathcal{CL}$ [64, 171] is another logical language based on dynamic logic that treats deontic operators as first-class citizens and has support for conditional obligations, permissions and prohibitions, as well as for (nested) contrary-to-duties and contrary-to-prohibitions. Singh proposed to capture normative concepts through the use of commitments [194, 196], which enables to model interactions in multiagent systems in a computationally realizable yet high-level manner. Different formalizations of commitments have been explored in the literature [221, 195, 152]. López y López et al. proposed a model of norms including components such as goals, addressees and beneficiaries (agents), context, exception, rewards and punishments, which is formalized in a language based on set-theory and first order logic [145, 146].

Approaches for norm representation can also be found in the research domain of business process management where the notion of norm is interchangeable with the notion of compliance rule. For example, in [172], compliance rules with time constraints are formalized using data-aware Petri nets. In [144], compliance rules are modeled by Petri nets with final markings and action labels that satisfy the monitor property. In both [16] and [143], compliance rules are expressed using Linear Temporal Logic. Goedertier and Vanthienen investigated the use of temporal deontic assignments on activities as a means to declaratively capture the control-flow semantics that resides in business regulations and business policies [81]. D'Aprile et al. proposed to use causal laws to represent norms in the action theory, and introduced a notion of commitment to model obligations [47].

As for constitutive norms, a widely accepted syntax has been devoted to the form of "counts-as" statements: "X counts as Y in context C" which was proposed in [187]. A counts-as statement is attached to a specific context since what counts-as what differs from situation to situation. The research of logical representation of constitutive norms was first found in [124]. Until recently, there was a leap in the formal analysis of constitutive norms, which were mainly promoted by the introduction of normative concepts in (multi-) agent system designs. For instance, based on the work of Jones and Sergot [124], Governatori et al. introduced a new logic for counts-as connections [83, 77], which allows for non-monotonic reasoning. Boella and van der Torre proposed to model both constitutive and regulative norms as conditional rules representing, respectively, the beliefs and goals of the normative system [28, 29]. Grossi et al. proposed a logical systematization of the notion of counts-as which is grounded on the intuition about forms of classification and is formally addressed by using modal logic techniques [94, 95].

2.5.2 Relation of Norms

When a set of norms is imposed to regulate the behavior of a system, it is possible that this set of norms is interrelated [180]. For example, a norm might be a reparation of the violation of another norm. From a system point of view, the interrelations between norms is important to determine the effects of the collective behavior of the agents. In deontic logic, a mostly studied relation between norms is the contrary-to-duty relation in which there is a primary obligation and what might be called a secondary obligation and the secondary obligation comes into effect when the primary obligation is violated [170]. An extensive study on different types of obligations (achievement, maintenance, punctual, (non-)preemptive) and the relations between obligations (violation conditions and reparation obligations) was given in [84, 88, 89]. López y López et al. introduced an interlocking relation between norms to model the situations that triggering a norm depends on past compliance with another norm [145, 146].

2.5.3 Contextualization of Norms

As it appears in human societies, norm specification may be extended to multiple levels of abstraction. Norms at a high abstraction level usually try to regulate a wide range of situations such that little maintenance is required over time. This is achieved by using abstract terms which are amenable of further interpretation at lower abstraction levels. In legal theory, such abstract terms are said to be open-textured [100], which can be interpreted by means of refining abstract norms to more concrete ones with respect to more specific contexts where the norms are applied. For example, a piece of local legislation is a refinement of the constitution of a country. In a law system, such a refinement process is realized by several layers of legislation and rule-giving from the constitutional level to the most concrete level of jurisprudence. This characteristic of human rule-giving system has attracted the attention of the researchers for the design of normative agent systems. For example, in [53], Dignum discussed how the connections between abstract norms and the concrete institutional protocols can be made. Later, Grossi et al. provided a formal analysis of this problem and proposed a formal framework by mixing the semantics of description logic with the idea of modeling contexts as sets of models [92, 93]. Aldewereld et al. proposed to use *counts-as* statements to relate the abstract concepts used in normative specifications to concrete ones in practice [8, 9].

2.6 Normative Multi-agent Systems

Multi-agent system (MAS) is often seen as a computerized system composed of interacting autonomous agents (as defined in Section 2.3) within an environment [216]. While individual agents are capable of controlling their own behavior in the pursuit of their individual goals, the overall objectives of MAS can be guaranteed by organizing and regulating the behavior of the individual agents and their interactions. Normative system is a widely used solution for this purpose, which is referred to any set of interacting agents (both human and computer system) whose behavior can be regarded as governed by norms. According to Jones and Sergot [123], "at the appropriate level of abstraction — law, computer systems, and many other kinds of organizational structure may be viewed as instances of normative systems".

In this section, we give an overview of the existing approaches of this kind in terms of two general categories: organization-oriented and institution-oriented, which result in what is often called normative multi-agent systems. Besides the normative frameworks in these two categories, there are several other works that do not have an organization or institution based architecture. For example, Boella and van der Torre [28] introduced a formal framework for the construction of normative multi-agent systems, based on Searle's notion of construction of social reality [188] and the input/output logic presented in [149]. Using the metaphor of normative systems as agents, they attribute mental attitudes to the normative system. Focused on the dynamic relations between agents in norm-governed societies, López y López and Luck [145] proposed a model for normative multi-agent systems which divides norms into three different classes that allow agents to identify the roles of other agents in a society and the limits of their responsibilities. These frameworks are useful in building computational agent systems, but they take a very low level approach to abstract agent interactions.

2.6.1 Organization-oriented Frameworks

Organizations in MAS can be understood as complex entities where a multitude of agents interact within a structured environment to achieve some global objectives [56]. In such a setting, coordination and control of agents are expanded in the sense that agent interaction is not designed in terms of the mental states of individual agents, but in terms of organizational concepts such as roles, norms and interaction protocols. Research on agent organizations translates solutions from human societies into electronic distributed computational mechanisms for the design and analysis of increasingly complex systems, and has gained terrain in agent community (e.g., the COIN workshop series [110]).

One of the earliest works is the Agent/Group/Role (AGR) model [65], which is based on organizational concepts such as groups, roles, and structures. This model describes an agent organization as a role-group structure imposed on agents and provides the basic foundational elements required in MASs to foster dynamic group formation and operation. Based on the AGR model, a multi-agent platform called MadKit is later presented in [97] for designing and simulating multi-agent systems. In AGR, agent interaction is captured as part of a role description while the normative aspects of interaction are not considered.

OperA [55] is a framework for the specification of MASs consisting of three interrelated models: (1) the *organizational* model which describes the organizational structure of a MAS by means of roles and interactions, (2) the *social* model which specifies how individual agents agree to enact roles, and (3) the *interaction* model which describes the possible interaction between agents. The organizational model reflects the organizational characteristics of a MAS in terms of four structures: social, interaction, normative and communicative. The *social* structure specifies objectives of the MAS, its roles and what kind of model governs coordination. The *interaction* structure describes interaction moments as scene scripts, representing a task that requires the coordinated action of several roles. Norms and regulations are specified in the *normative* structure. Finally, the *communicative* structure specifies the ontologies for describing domain concepts and communication illocutions. Based on the OperA framework, a tool called OperettA was developed to build and analyze organizational models[10]. Though OperA provides an efficient model for designing MAS, it does not consider agent individuality and the design mainly stays at a single level without taking into account the contextual aspect. Moreover, OperA does not provide an operationalization for organizational design.

HARMON*IA* [208] is a framework to model electronic organizations at four levels of description: (1) *abstract level* where the statues of the organizations are defined in a high level of abstraction along with the first abstract norms, (2) *concrete level* where abstract norms are iteratively concretized into more concrete norms, and the policies of the organization are also defined, (3) *rule level* where concrete norms and policies are fully refined, linking the norms with the ways to ensure them, and (4) *procedure level* where all rules and policies are translated in a computationally efficient implementation easy to be used by agents. This division of levels remarks the distinction between the normative dimension of the electronic organization (what is acceptable or unacceptable by means of obligations, permissions and rights) and the descriptive dimension (how agents should behave in order to meet the norms). HARMON*IA* provides a flexible structure of norm specification from abstract to concrete but it ignores the characteristics of agents in organizational design.

Moise+ [111] is an organizational modeling language that decomposes the specification of organization into structural, functional, and deontic dimensions. The structural dimension specifies the roles, groups and links of the organization. The functional dimension specifies how the global goals are decomposed, grouped and distributed to the agents. The deontic dimension binds the other two dimensions through the roles' permissions and obligations for missions. Together with two other technologies CArtAgO [178] and Jason [30], a framework for multi-agent programming called JaCaMo was proposed, in which Moise+ is used for programming multi-agent organizations, CArtAgO is used for programming environment artifacts, and Jason is used for programming autonomous agents. Moise+ provides a powerful tool for the engineering of MAS by combining with other models and methods. However, it does not provide a mechanism of incorporating agent individuality in the evaluation of organizational design. Moreover, Moise+ does not have the constructs for compliance evaluation.

ROMAS [74] is a methodology for the analysis and design of normative open MAS, with its roots in GORMAS [14] and OperA. It consists of four views: (1) the organizational view specifies a system in terms of its global purposes, its relationship with the environment, the division of roles, (2) the internal view specifies each entity (organizations, agents and roles) of the system in terms of its believes, objectives as well as the task and services it implements, (3) the contractTemplate view specifies contract templates which will be used at runtime as a starting point for negotiating contracts. The activity view specifies the sequence of activities in which a task or service implementation is decomposed. The engineering of MAS in ROMAS is supported by a CASE tool called EMFGormas [75]. In ROMAS, normative specification covers constitutive, regulative and procedural aspects, but the interrelations between norms are not explicitly represented. Moreover, ROMAS does not provide a mechanism to incorporate agent autonomy in the evaluation of organizational design.

2OPL is a programming language that supports the implementation of normative multi-agent organizations [49, 203]. It considers an organization as a software entity that exogenously coordinates the interaction between agents and their shared environment. 2OPL provides programming constructs to specify (1) the initial state of an organization, (2) the effects of agents' actions in the shared environment, and (3) the applicable norms and sanctions. Norms in 2OPL can be either enforced by means of sanctions or regimented. 2OPL has the advantage that it comes with a complete operational semantics such that normative organization programs can be formally analyzed by means of verification techniques [15]. However, organizational design in 2OPL is at a very low operational level and requires a full description of the organizational states.

Focused on the normative aspects of organizations, Singh [196] proposed an approach for governing sociotechnical systems that is computationally realized and deals well with complexity and dynamism. The conceptual model of a socialtechnical system is centered on the notion of *principals* representing the participants in a system which can be either an individual or an organization. By allowing principals to choose their own policies, participants' autonomy is incorporated. In this approach, *organizations* are structured through the specification of *roles*, which codifies a set of related interactions and can be enacted by the principals in an organization. An organization can be considered as a principal and can thus participate in another organization by playing a role therein. The regulation of the interactions of the principals is realized by the specification of *norms* with a rich set of constructs and types. The proposed system can change according to stakeholders' needs or physical circumstances via adapting participants' policies, expanding role-enacting organizations and changing compliance rules. This approach has some overlap with the one presented in this dissertation such as recursive modeling of organizations and the consideration of individual autonomy. However, there has not been works based on this approach regarding modeling interrelations between norms, evaluating degrees of compliance and detecting norm conflicts.

Besides the agent organizational frameworks described above, there are a large number of MAS methodologies proposed by different research groups, such as SODA [164], INGENIAS [168], MaSE [51], Tropos [34], Gaia [223], ASPECS [46], etc. Although these methodologies adopted organizational concepts such as role in the analysis and design phase of MAS development, they are mostly based on a centralized approach to system design, in which agents are designed to fulfill the roles, and do not support specification of open systems and incorporate agent personalities in role enactment.

2.6.2 Institution-oriented Frameworks

Institution is another human-society motivated concept that has been applied to the design of MAS. According to Ostrom [165], institutions can be seen as the prescriptions that humans use to organize all forms of repetitive and structured interactions within various social settings. Inspired by such an organizational structure, several frameworks have been proposed for the specification of electronic institutions which are often seen as the model of a (human) institution through the specification of its norms in some suitable formalisms [161, 208].

The result of an early work is ISLANDER [179] for the specification and verification of agent mediated electronic institutions. It consists of four basic elements: (1) *dialogic framework* which defines the valid illocutions that agents can exchange and the participant roles and their relationship, (2) *scenes* which define agent interactions through the model of possible dialogic interactions between roles, (3) *performative structure* which defines collections of multiple, concurrent scenes, and (4) *norms* which define obligations for participating agents. Furthermore, Esteva et al. proposed an infrastructure AMELI [58] to support the execution of electronic institutions that mediates agents' interactions while enforcing institutional rules. ISLANDER provides a comprehensive framework for the design of institutions but it does not allow agents to violate the norms.

InstAL [43] is a normative framework with a formal mathematical model to specify, verify and reason about the norms (obligations, permissions, powers) that govern an open distributed system. It opts for an event-driven institutional approach, i.e., the norms are expressed over the events/actions of the participants rather than the normative state. The premise of the model is that events trigger the creation of institutional fluents based on two types of rules: (1) *causal rules* that describes which fluents are initiated and terminated by the performance of an action, and (2) *generation rules* that describe when the performance of one event counts-as or generate another. The formal model and semantics is translated in an equivalent logic program under the answer set semantics. To support normative design, an inductive logic programming system was developed [45]. InstAL provides a more concise model at the expense of the designer having to hard-code the normative constructs.

OCeAN [70] is a framework that can be used to specify electronic institutions. The fundamental concepts of this model are: (1) an ontology for the definition of the concepts used in the communication and in the specification of the rules of the interaction, (2) the definition of the events, the actions, and the institutional actions and events that may happen or can be used in the interaction among agents, (3) the definition of the roles that the agents may play, (4) an agent communication language (ACL) for the exchange of messages, (5) the definition of the institutional powers for the actual performance of institutional actions, and (6) a set of norms for the specification of obligations, prohibitions, and permissions. This model has been formalized using the Discrete Event Calculus, and recently Semantic Web technologies were applied for specifying and monitoring obligations [69].

The three frameworks ISLANDER, InstAL and OCeAN take their inspiration from institutions where the norms are expressed at a single level of the actions of the participants. Such a low level approach makes it difficult to apply them for the design of large-scale systems. Moreover, none of them provides mechanisms of incorporating agent individualities in the evaluation of institutional design.

2.7 Norm Compliance and Conflict

Compliance checking and conflict detection are two important subjects in the research of normative systems. Compliance checking concerns the determination of whether agents' behavior is in accordance with a set of norms, and conflict detection concerns the determination of whether the set of norms itself is consistent. Both of them have attracted a lot of attention in the research of normative multi-agent systems.

An early work was presented by Venkatraman and Singh, which developed an approach for testing whether the behavior of an agent complies with a commitment protocol formalized in temporal logic [209]. The link between commitments and normative concepts was investigated by Singh [194]. With the aim of checking compliance of interaction protocols against norm specification, Aldewereld et al. introduced a formal method based on program verification [11]. Herzig et al. presented a model checker [127] for reasoning about compliance in normative systems based on the logical system presented in [5]. With a similar normative structure, Bulling et al. provided a formal analysis of monitor and studied the relations between monitors and norms to be monitored [37]; Herzig et al. proposed a dynamic logic to reason about abilities and permissions of agents that can be used to find executable and authorized procedures that ensure the achievement of some desired results [103]. These approaches do not have a high level specification of normative concepts but specify a normative system as a kind of transition system, which often requires a full description of system states. As for norm conflicts, an early work is presented in [183], in which the concept of normative conflict is formally analyzed and two approaches of reasoning with normative conflicts are discussed. Vasconcelos et al. applied first-order unification to discover overlapping substitutions to the variables of laws/norms in which legal/norm conflicts may occur [207, 206]. Targeting distributed management of norms, Gaertner et al. proposed a normative model NS based on the propagation of normative positions (obligations, prohibitions, permissions) as consequences of agents' actions, and realized conflict detection by providing a mapping of NSs into Colored Petri Nets [72]. Focused on normative conflicts in electronic contracts, Giannikis and Daskalopulu presented a set of primitive conflict patterns and proposed the representation of e-contracts in default logic to facilitate conflict detection [79]. Based on the institutional model InstAL [43], Li et al. proposed a computational model for detecting norm conflicts given traces of agent actions by means of Answer Set Programming [140]. Focused on identifying conflicts between obligations in dynamic settings, Tosatto et al. introduced a new semantics for the obligations to identify the necessary and sufficient conditions to detect conflicting obligations [204].

Another line of relevant work comes from the research domain of business process management, which is promoted by the advent of stricter legal demands, industrial best practices and prescriptive enterprise architectures. According to [128], there are mainly two types of approaches: forward compliance checking and backward compliance checking, in which the former targets the verification of rules during design time or execution time while the latter can detect that non-compliant behavior has taken place by looking at the execution history of business process.

Adopting a forward compliance checking paradigm, Lohmann proposed to automatically construct business process models that are compliant by design based on an existing artifact-centric framework [144], in which Petri Nets are used to model artifact life cycles, inter artifact dependencies, compliance rules, and a role-based access control. Awad et al. introduced an approach to synthesize business process templates out of a set of compliance rules expressed in linear temporal logic [16]. The approach also shows how analysis is conducted if the compliance rules are inconsistent. Liu et al. proposed to use Pi-Calculus to formalize process models and linear temporal logic to represent compliance rules such that process models can be verified against these compliance rules by means of model-checking techniques [143]. Goedertier and Vanthienen investigated the use of temporal deontic assignments on activities as a means to declaratively capture the control-flow semantics that resides in business regulations and business policies [81]. D'Aprile et al. proposed to specify business processes as action domains and norms with the notion of commitments, and verify compliance of a business process based on temporal answer sets [47]. In order to reason about violations of obligations in contracts, Governatori and Milosevic proposed a formal contract language called FCL [85] based on which approaches for business process compliance were intensively explored [182, 87, 89]. This line of research based on FCL provides a powerful way of modeling different kinds of norms and investigated the reparation relations between norms. The difference between their approaches and our work is that we explicitly consider the concepts of agents and roles in normative specifications which facilitates compliance checking from an organizational perspective.

Following a backward compliance checking paradigm, process mining techniques [1, 2] are widely used to find commonalities and discrepancies between the modeled behavior and the observed behavior. For example, based on the notion of alignment, a number of approaches have been proposed to check compliance by aligning the observed behavior described in event logs with the expected behavior specified in compliance rules concerning control-flow, data-flow and organizational aspects [172, 136, 137]. Aiming at alleviating the laborious work of auditors, van der Aalst et al. proposed an online auditing tool, in which business rules are translated in a straightforward way into queries that can be executed against the database consisting of three types of data (run time data, de jure models, and de facto models) [4]. Based on a complete set of enriched process event data, Caron et al. presented a comprehensive rule-based compliance checking approach [40]. The approach provides a business rule taxonomy for classifying the generic rule patterns according to their process mining perspective and their rule restriction focus. In [157], two declarative process mining tools are introduced. The first is called SCIFF Checker which classifies execution traces as compliant or non-compliant with declarative business constraints expressed in a textual pseudo-natural syntax. The second is called DecMiner which employs inductive logic programming techniques to discover a declarative constraint-based specification based on a set of execution traces labeled as compliant or non-compliant (according to the results from the SCIFF Checker).

In this research, we take a backward compliance checking approach to verify whether executions of business processes are complying with certain normative constraints. Our work is based on the formal theories of normative systems, which provide a precise and unambiguous specification of the normative constraints that the entities are subject to in the context where the normative system applies [154, 13]. Importantly, we take an integrated view on compliance checking, i.e., formalizing and analyzing the impact of the interrelations between compliance rules, which has not been explicitly explored in the aforementioned compliance checking approaches.

2.8 Colored Petri Nets

Colored Petri Nets (CPNs) [112] is a discrete-event modeling language combing the capabilities of Petri Nets with a high-level programming language standard ML [155]. Petri Nets, originally introduced by Petri in 1962 [169], provides the foundation of the graphical notation and the basic primitives for modeling concurrency, communication and synchronization [159]. CPNs are typically applied in domains such as communication protocols, data networks, distributed algorithms, and embedded systems. They are also applicable more generally for modeling systems where concurrency and communication are key characteristics, e.g., business processes and workflows, manufacturing systems, and agent systems.[1]

2.8.1 Basics

Unlike classical Petri nets, tokens in CPNs are distinguishable, i.e., tokens in CPNs may have different colors or data types and carry data attributes that characterize the entities the tokens represent. As an example, we consider a customs inspection scenario in international trade. Loaded with goods, carriers with or without AEO certificate arrive at the customs building and the customs carries out inspection accordingly. Figure 2.1 shows a simple CPN model. In the rest of this section, we explain this example in detail and introduce the CPN formalism.

A CPN is a directed graph consisting of two types of nodes, called *places* and *transitions*, where *arcs* are either from a place to a transition or from a transition to a place. A place serves as a placeholder for the entities in the system being modeled and is graphically represented as a circle or ellipse. There are four places in Figure 2.1, respectively labeled with *Loaded*, *Waiting*, *Free* and *Done*. A transition represents an action that can occur in the system and is graphically indicated by a rectangle. There are two transitions in Figure 2.1, respectively labeled with *Arrive* and *Inspect*. Arcs describe how data flows when transitions are executed. Places having arcs which connect them to a transition, are called *input places* of the transition. Similarly, the places connected to a transition

[1] www.cs.au.dk/CPnets/industrialex

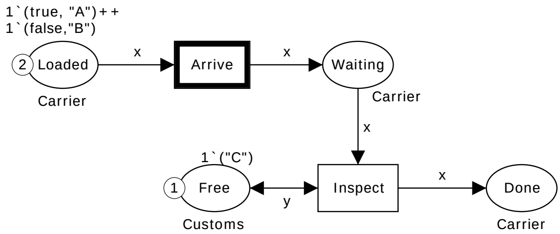


Figure 2.1: *Simple CPN model of customs inspection*

by arcs going from the transition are called *output places* of the transition. In particular, a place without any input transitions is called a *source place*, and a place without any output transition is called a *sink place*. For example, in Figure 2.1, *Loaded* is a source place and *Done* is a sink place.

Each place is associated with a type or a color set that determines the kind of tokens the place may contain. In Figure 2.1, the places *Loaded*, *Waiting* and *Done* are of type Carrier and the place *Free* is of type Customs. This indicates that we can only have tokens with the type of Customs in place *Free* and only tokens with the type of Carrier in the remaining places. Types or color sets are explicitly declared in CPNs by the language CPN-ML. In the example, we have the following declarations of the color sets:

colset AEO = bool;
colset Name = string;
colset Carrier = product AEO * Name;
colset Customs = Name;

The first line indicates that the color set AEO corresponds to the simple type *bool* (boolean). The second line specifies that the color set Name corresponds to the simple type *string*. The third line specifies that the color set Carrier is a combination of the types AEO and Name using the color set constructor *product*. The fourth line indicates that the color set Customs is defined by the type Name. Declarations allow us to give different names to simple types such that the model is more readable.

Besides a type, each place has a marking to indicate its state, which is defined as a multiset of values over the type of the place. A multiset is similar as an ordinary set except that the same element can occur multiple times. A token is an element of such a marking, i.e., it has a value and resides in a place. In Figure 2.1, a marking of a place is shown in a circle and a piece of text near the place, such as the circle containing 1 and the text 1‘(“C”) on place *Free*. The number in the circle represents the total number of tokens in the place and the text is a textual representation of the multiset of tokens. In the example, place *Free* contains exactly one token and the marking is written 1‘(“C”). The use of a backwards apostrophe (‘) is to separate the value of a token and the count of how many tokens with that value is part of the marking. If a marking consists of tokens with different values, we separate them with two pluses (++). For example, place *Loaded* contains a multiset of tokens written 1‘(true,“A”)++1‘(false,“B”), representing a marking containing two tokens, one with value (true,“A”) and the other with value (true,“B”). Places without a marking contain an empty multiset which is not shown explicitly. An assignment of markings to all the places of a CPN is called a marking of the CPN, representing the state of the model.

Arcs connecting to a transition can be categorized into *input arcs* and *output arcs*, in which an input arc connects a place to a transition and an output arc connects a transition to a place. An arc has an inscription or expression which may contain one or more free variables. In Figure 2.1, transition *Arrive* has an input arc from place *Loaded* with expression x and an output arc with the same expression. Variables in the expressions must be declared to be of a certain type. In the example, two variables are declared.

var x: Carrier;
var y: Customs;

Variable x is of type Carrier and variable y is of type Customs. An arc expression must have the same type as the type of the place it is connected to or a multiset of the place type.

A transition has a set of variables, i.e., the ones occurring on all the arcs connecting to it. Each of these variables can be assigned a value from the set represented of its type. For example, the variable x can be assigned the value (true,“A”) and (false,“B”). A transition along with an assignment of each of its variables is referred to as a *binding element* (or binding for short), which is denoted by the name of the transition and a list of assignment to all its variables in braces. In our example, there are binding elements Arrive⟨x=(true,“A”)⟩ and Arrive⟨x=(false,“B”)⟩. Given a binding element, we can evaluate the expressions on all arcs connecting to the transition. For example, given the binding element Arrive⟨x=(true,“A”)⟩ in Figure 2.1, the expression x on the input arc from *Loaded* and the output arc to *Waiting* evaluates to (true,“A”).

Given a CPN with a marking and a binding, the binding is considered to be enabled if all input places contain at least the tokens specified by the evaluation of the expression on the corresponding input arcs in the binding. For example, the binding element Arrive⟨x=(true,“A”)⟩ is enabled as the expression on the only input arc to the transition can evaluate to (true,“A”). A transition is enabled in a marking if there exists at least one binding element which is enabled in the marking. In Figure 2.1, there is one enabled transition, *Arrive*, indicated by the enabled binding elements Arrive⟨x=(true,“A”)⟩ and Arrive⟨x=(false,“B”)⟩. Enabled transitions are marked by a bold outline. Note that whenever a marking has multiple enabled bindings, one of these bindings is picked.

If a binding element or a transition is enabled, it can occur or be fired. This results in consuming all tokens from input places corresponding to the evaluations of the expressions on input arcs and producing new tokens on output places corresponding to the evaluations of the expressions on output arcs. In our example, when Arrive⟨x=(true,"A")⟩ occurs, it consumes the token with value (true,"A") from place *Loaded* and produces a token with value (true,"A") in place *Waiting* according to the arc expression x, leading to a situation as shown in Figure 2.2. Compared with Figure 2.1, only the marking has changed while the net structure remains the same. Transition *Arrive* remains enabled in this marking as it is enabled by the binding Arrive⟨x=(false,"B")⟩. Transition *Inspect* is now enabled with a binding ⟨x=(true,"A"),y=("C")⟩.

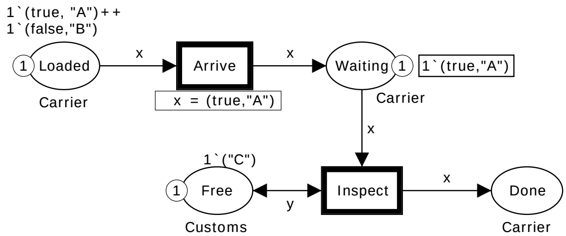


Figure 2.2: *Customs inspection model after the occurrence of binding element* $Arrive\langle i = (true, "A'')\rangle$

The marking of a model before we start simulation is called the initial marking. For example, in Figure 2.1, place *Loaded* contains two tokens with value (true, "A") and (false, "B"), and place *Free* contains one token with value ("C"). After executing the binding Arrive⟨x=(true,"A")⟩, the new marking is depicted in Figure 2.2, which is referred to as the current marking. Execution of a binding element is called a *step*.

2.8.2 Definition

In this section, we give a formal description of a CPN model and the related concepts excerpted from [112].

Definition 2.1 (CPN) *A CPN is a tuple (P, T, A, Σ, V, C, E, I) where*

- P *is a finite set of places,*
- T *is a finite set of transitions such that* $P \cap T = \emptyset$*,*
- $A \subseteq P \times T \cup T \times P$ *is a set of directed arcs,*
- Σ *is a finite set of non-empty color sets (data types),*
- V *is a finite set of typed variables such that* $Type[v] \in \Sigma$ *for all variables* $v \in V$*,*

- $C : P \rightarrow \Sigma$ *is a color set function that assigns a color set to each place,*
- $E : A \rightarrow EXPR_V$ *is an arc expression function that assigns an arc expression to each arc* $a \in A$ *such that* $Type[E(a)] = C(p)_{MS}$*, where* p *is the place connected to the arc* a *and* MS *indicates* $C(p)_{MS}$ *is a multiset,*[2]
- $I : P \rightarrow EXPR_\emptyset$ *is an initialization function that assigns a closed expression to each place* p *such that* $Type[I(p)] = C(p)_{MS}$.

Definition 2.2 *Given a* $CPN = (P, T, A, \Sigma, V, C, E, I)$*, the following concepts are defined:*

1. *A marking is a function* M *that maps each place* $p \in P$ *into a multiset of tokens* $M(p) \in C(p)_{MS}$,
2. *The initial marking* M_0 *is defined by* $M_0(p) = I(p)\langle\rangle$ *for all* $p \in P$,
3. *The variables of a transition* $t \in T$ *are denoted* $Var(t) \subseteq V$*, including the free variables appearing in the arc expressions of the arcs connected to* t,
4. *A binding of a transition* $t \in T$ *is a function* b *that maps each variable* $v \in Var(t)$ *into a value* $b(v) \in Type[v]$*, and the set of all bindings for a transition* t *is denoted* $B(t)$,
5. *A binding element is a pair* (t, b) *such that* $t \in T$ *and* $b \in B(t)$*, and the set of all binding elements* $BE(t)$ *for a transition* t *is defined by* $BE(t) = \{(t, b) | b \in B(t)\}$,
6. *A step* $Y \in BE_{MS}$ *is a non-empty, finite multiset of binding elements.*

For a binding element (t, b), $E(a)\langle b\rangle$ denotes the result of evaluating the arc expression $E(a)$ of the arc a in the binding b. For a given place p, $E(p, t)$ denotes the arc expression on the input arc from p to t, and $E(t, p)$ denotes the arc expression on the output arc from t to p. The dynamic behavior of a system is reflected by the changes of the marking of a CPN, as described in Definition 2.3.

Definition 2.3 (Enabling and occurrence of steps) *A step* $Y \in BE_{MS}$ *is enabled in a marking* M *iff* $\forall p \in P : \overset{++}{MS} \sum_{(t,b)\in Y} E(p, t)\langle b\rangle \ll= M(p)$
where the symbol ++ to the upper left of the summation symbol is used to indicate that it is a sum of multisets, and the symbol $\ll=$ *is used to indicate that a multiset is smaller than or equal to another multiset.*

When the step Y *is enabled in the marking* M*, it may occur, leading to another marking* M' *defined by:*

[2] a multiset is like an ordinary set (i.e., the order of elements does not matter), but the same element can occur multiple times.

$$\forall\, p \in P : M'(p) = (M(p) \underset{MS}{\overset{++}{--}} \sum_{(t,b)\in Y} E(p,t)\langle b\rangle) \underset{MS}{\overset{++}{++}} \sum_{(t,b)\in Y} E(t,p)\langle b\rangle$$

where the symbol $--$ *is used to indicate the subtraction of two multisets.*

A finite sequence of steps and markings is called an *occurrence sequence*. When there exists an occurrence sequence starting from a marking M which leads to another marking M', we say M' is reachable from M, denoted as $\mathscr{R}(M, M')$.

2.8.3 Hierarchical CPNs

The CPN formalism we have introduced is sufficient to model the behavior of complex systems, but it captures the system behavior in one single net. When systems become larger, the models would be too large to get a clear overview. In this section, we introduce the formalism of Hierarchical CPNs to facilitate the modeling of large and complex systems, the idea of which is to decompose the model of a system into a set of modules. A module is a CPN with a set of interface places, which can be used to describe the internal structure of a substitution transition.

Continuing with the example in Figure 2.1, now suppose the customs inspection is different between carriers with and without AEO certificates. In this case, we just want to focus on the elaboration of the inspection part of the original model. To do this, we take the CPN model in Figure 2.1 as a top-level module and replace transition *Inspection* with a substitution transition which refers to a sub-module that gives a detailed description of the inspection process. Figure 2.3 shows the CPN modeling the top-level module. The double-lined rectangle with the tag Inspection under it denotes the sub-module. The places connecting to the substitution transition are called *socket places*, serving as the interface between the modules.

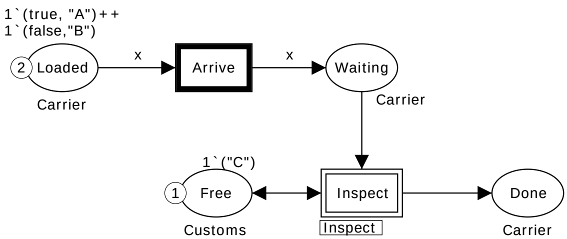


Figure 2.3: *CPN model of the customs inspection top module*

A module is a CPN with two kinds of transitions: *elementary transitions* and *substitution transitions*. An elementary transition is a normal transition as defined in the previous section. A substitution transition refers to a module and it may have internal states and does not need to consume and produce tokens in one atomic action. A module may contain multiple substitution transitions which refer to other modules that in turn may contain substitution transitions referring to other modules. There can be an arbitrary number of levels as long as no cycles are introduced as this would result into an infinitely large model.

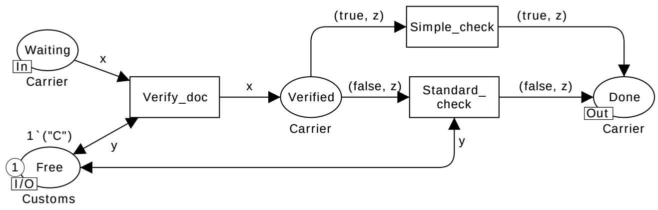


Figure 2.4: *CPN model of the inspection module*

The top-level module in Figure 2.3 has one substitution transition *Inspection* referring to module Inspection, indicated by the tag under it. Figure 2.4 shows the CPN model of the module *Inspection* which has three interface places: *Waiting*, *Free*, and *Done*. Such places are called *port places*. In oder to relate a substitution transition to the CPN module it refers to, we need to relate each socket place of the substitution transition to a port place of the CPN module. As a result, each pairs of socket place and port place is semantically merged into one place. For example, socket place *Waiting* in Figure 2.3 is merged with the port place with the same label in Figure 2.4. In particular, each port place has a type, indicating its type of connection, i.e., whether tokens are coming in or going out of the port place. For example, in Figure 2.4, places *Waiting*, *Done* and *Free* are respectively tagged with In, Out, and I/O (input and output).

In the module Inspection, there are three transitions *Verify_doc*, *Simple_check* and *Standard_check*, and one place *Verified* which is of type Carrier, besides the port places. The inspection process starts with a document verification and continues with a check procedure depending on whether a carrier is AEO certified. For those with an AEO certificate, a simple check will be carried out by the customs. Otherwise, a standard check will apply. Such a difference is indicated by the inscriptions respectively defined on the input arcs from place *Verified* to transition *Simple_check* and *Standard_check* in which z is a variable of type Name. That is, tokens with AEO being true will be consumed by transition *Simple_check* and that with AEO being false by transition *Standard_check*.

Based on the description above, we give a formal description of a Hierarchical CPN model and the related concepts excerpted from [112].

Definition 2.4 (CPN module) *A CPN Module is a tuple* $CPN_M = (CPN, T_{sub}, P_{port}, PT)$ *where*

(1) $CPN = (P, T, A, \Sigma, V, C, G, E, I)$,

(2) $T_{sub} \subseteq T$ *is a set of substitution transitions,*

(3) $P_{port} \subseteq P$ *is a set of port places, and*

(4) PT: $P_{port} \rightarrow$ {IN, OUT, I/O} *is a port type function that assigns a port type to each port place.*

Each module extends a CPN as defined in Definition 2.1. Specifically, a subset of the transitions T_{sub} in the CPN are tagged as substitution transitions and a subset of the places P_{port} in the CPN are tagged as port places. An assignment of types to the port places indicates whether tokens are coming in or going out of the port places.

Definition 2.5 (Hierarchical CPN) *A hierarchical CPN is a tuple* $CPN_H = (S, SM, PS)$ *where*

(1) S *is a finite set of modules,*

(2) $SM : T_{sub} \rightarrow S$ *is a submodule function that assigns a submodule to each substitution transition, and*

(3) PS *is a port-socket relation function that assigns a port-socket relation* $PS(t) \subseteq P_{sock}(t) \times P_{port}^{SM(t)}$ *to each* $t \in T_{sub}$.

SM specifies the mapping between substitution transitions and corresponding CPN modules. Be noted that the mapping relation should not incur cycles in the hierarchical CPN. As to the interface places, the input places of substitution transitions are called input sockets and the output places are called output sockets. A *port-socket relation* PS relates the port places of the submodule to the socket places of the substitution transition. When a port and a socket are related, the two places constitute two different views of a single place, which means that they always share the same marking.

Each hierarchical CPN builds a module hierarchy, i.e., a directed graph in which each node represents a CPN module and each labeled arc links a substitution transition to its corresponding module, as formalized in Definition 2.6.

Definition 2.6 (Module hierarchy) *The module hierarchy for a hierarchical CPN is a directed graph* $MH = (N_{MH}, A_{MH})$*, where*

(1) $N_{MH} = S$ *is the set of nodes,*

(2) $A_{MH} = \{(s_1, t, s_2) \in N_{MH} \times T_{sub} \times N_{MH} | t \in T_{sub}^{s_1} \wedge s_2 = SM(t)\}$ *is the set of arcs.*

The roots of MH are called *prime modules*. An *instance* of a module (denoted as s^*) corresponds to a path in the module hierarchy leading from a prime module to the module s. When a module instance is created, a *place instance* (denoted as p^*) and *transition instance* (denoted as t^*) are created for each place and transition in the module. A *compound place* (denoted as p_{cp}) is a set of place instances related via port-socket relations. Place instances belonging to the same compound place are equivalent in the sense that they always have identical markings. When a place instance is not equivalent to any other place instance, it will form its own compound place. A marking M for a hierarchical CPN model is a function that maps each compound place into a multiset of tokens representing the marking of the place instances belonging to the compound place. The concepts of variables of transitions, bindings, binding elements, and steps are defined in a similar way to that for non-hierarchical CPNs except that transition instances are considered instead of transitions and bindings are only defined for non-substitution transitions.

2.8.4 Tool Support

The practical application of modeling and validation relies heavily on the existence of computer tool support. Developed originally by the research group at Aarhus University, **CPN Tools** [113] is a tool for the editing, simulation, state space analysis, and performance analysis of CPN models.

CPN Tools supports two types of simulation: interactive and automatic. In an interactive simulation, the user is in complete control and determines the individual steps in the simulation, by selecting among the enabled transitions in the current state. CPN tools will show the resulting marking of executing a selected step in its graphical representation of the CPN model. In an automatic simulation, CPN tools will make random choices among the enabled bindings in the states encountered and will stop on the conditions set by the users, e.g., stop after a certain number of steps, stop when there is no enabled transitions, stop when an expression evaluates to true, or simply a breakpoint.

2.9 Agent Systems and Petri Nets

With the subjects of agent systems and Petri nets respectively attracting tremendous attention, a lot of research has also been done on applying Petri Nets and their variants to the modeling and verification of (multi-) agent systems. For example, the research group in the University of Hamburg proposed a multi-agent architecture MULAN [131, 39] which is built on reference nets [134] and can be executed in a Petri net editor and simulator RENEW [135]. Furthermore, a formal organization model for agent systems was presented [130, 132], and an organization editor called OREDI was developed to enable editing organization models and deploying such models as MULAN systems. This Petri net based model directly integrates organizational concepts such as roles, teams etc., allowing an alignment of business and IT. In addition, the automated mapping of the proposed models to multi-agent systems is supported. Regarding multi-agent systems as discrete-event dynamic systems, Celaya et al. investigated how Petri nets can be used as a modeling tool to assess the interaction properties of the multi-agent systems

[42]. This line of research provides a powerful link between the development of Petri Nets and the modeling and analysis of multi-agent systems. However, none of them consider the normative aspect of agent systems. An early work that exploits the use of Petri Nets for the analysis of normative behavior can be found in [176]. Targeting regulated multi-agent systems, Gaertner et al. proposed a normative structure NS and a mapping to Colored Petri Nets for the detection of norm conflicts [72]. Though providing useful formalisms, these works are not intended for the analysis of agent behavior in an organizational setting.

2.10 Conclusions

In this chapter, we provided an overview of the state-of-the-art theories and methodologies in the research areas related to the topic of this dissertation and justify the choices made for this research.

We have shown that, inspired by the working mechanisms of human organizations and institutions (studied in, for example, organization and institution theory), normative systems can be an answer to the problem of coordination and cooperation in multi-agent systems where autonomous agents interact together within an environment for some collective goals. The resemblance between such agent systems and organizations in human societies indicates the fact that, being able to ensure the achievement of the collective goals, both need adequate levels of control since with various interests and needs participants may (un)intentionally deviate from the collective goals for personal gain. Therefore, control mechanisms are needed to be employed to guarantee a certain level of compliance.

Organizational control indicates limiting the way that the participants may perform. On the other hand, individual participants have their personal interests and needs, and it is essential to incorporate such individual characteristics into the evaluation of different organizational structures to find a balance between the conformity desired by the organization and the autonomy desired by the participants. Inspired by human decision making, preference reasoning can be an answer to represent the desires of agents in selecting plans to achieve their individual objectives. This provides a potential approach to capture the autonomy desired by participants in performing actions to achieve organizational objectives. In the following chapters, we illustrate in detail about how these aspects of control and autonomy are considered in the design and evaluation of agent interactions.

Part I

OperA+ Framework

Chapter 3

Overview

This Chapter has the following contribution:

- an agent-based framework for organizational modeling.

In this chapter, we present the outline of the organizational framework OperA+.[1] This framework is motivated by the realization that participants in organizations are autonomous entities who have individual interests and control over their own actions, and organizational compliance requires careful design and evaluation. The main focus of this framework is on designing and analyzing organizational models that capture both the compliance requirements of organizational interactions and the characteristics of individual participants. A model based on the OperA+ framework is intended to regulate and guide the behavior of participating agents in order to ensure the achievement of some collective objectives in a compliant way. More importantly, OperA+ also provides mechanisms to evaluate the effectiveness of agent interactions considering both system constraints and individual preferences.

The contributions of OperA+ are

- a framework of modeling organizations at multiple levels of abstraction, which enables an analysis of organizational behavior from abstract values to implementation details.

This chapter is based on our contributions to the 12th International Conference on Autonomous Agents and MultiAgent Systems, pp. 1435–1436, 2014 [114]; the 2012 IEEE/WIC/ACM International Conference on Intelligent Agent Technology (IAT), 196–203 [121]; the 13th International Workshop on Coordination, Organisations, Institutions and Norms @ IAT 2011, pp. 58-74, 2011 [119].

[1] The name is an extension of **OperA** which stands for "**O**rganizations **per** **A**gents".

- a mechanism of reflecting the impact of environment characteristics on the structuring of organizational behavior, which facilitates the representation of compliance requirements in changing environments.
- a mechanism of integrating normative constraints with individual preferences in the evaluation of agent interactions, which enables a balance between organizational control and individual autonomy.

This chapter is organized as follows. Section 3.1 provides the motivation behind the proposal of OperA+. Section 3.2 introduces the basis of OperA+ and the reasons of such a choice. Thereafter, we illustrate the architecture of the OperA+ framework in Section 3.3. In Section 3.4, we discuss about the essential features of OperA+ models. Finally, we conclude this chapter and indicate directions for future work.

3.1 Motivation

As stated in Chapter 1, compliance is important for organizations to ensure the success of the participants' interactions. Organizations are urged to take measures to, identify the compliance requirements that are relevant to the regulation of their behavior, and ensure that their actual behavior is in accordance with the compliance requirements. To this end, there is a need for means that can help to formalize and evaluate organizational compliance. First of all, such a means should be able to capture the essential properties of the organizations which are intended to coordinate the behavior of multiple participants through explicit regulation structures such that organizational objectives can be realized in a compliant way. The participants can be humans, software, or organizations like a business company, a university, etc., and they are autonomous entities that can act independently and control their own behavior.

Moreover, according to the four research questions listed in Chapter 1, OperA+ should be able to meet the following requirements.

1. provide a formal representation of organizational compliance, taking into account various regulation sources and the environmental characteristics.
2. provide a mechanism to evaluate whether the behavior of the participants is complying with the compliance requirements of organizations,
3. provide a mechanism to deal with the combination of different regulation sources with respect to consistency.
4. provide a way to represent the preferences of individual participants and incorporate such preferences in the evaluation of organizational compliance.

From the perspective of sociology and organization theory, organizations can be understood as complex entities where a multitude of agents interact within a structured environment for some global objectives. Since the 80's, Multi-Agent Systems (MAS) have often been defined as organizations or societies of agents whose interactions are coordinated for the achievement of some collective goals [66], and an agent in this context is usually seen as an encapsulated computer system that is capable of flexible, autonomous action in a situated environment in order to meet its design objectives [217]. Agent organizations are committed to the integration of organizational control with individual intelligence, the dynamic adaptation to environmental changes, and rely for a great extent on the notion of openness and heterogeneity of MAS [56]. MAS design frameworks provide structures for developing organizational models, which also provide insight on how natural organizations may be augmented or enhanced [110].

Agent organizational models are a promising solution to the problem of organizational compliance. On the one hand, the agent paradigm provides a suitable representation of the autonomous participants in organizations. On the other hand, the MAS organization structures provide the constructs by which organizational compliance can be expressed and analyzed. More importantly, the well-developed theories and techniques in MAS organization design can be an asset for the study and improvement of organizational compliance.

As shown in Chapter 2, there has been a large number of agent organizational frameworks and methodologies that have been proposed for the design and analysis of MAS. These frameworks and methodologies lay a good foundation for the research of agent organizational modeling.

3.2 OperA as a Cornerstone

The OperA+ framework described in this chapter is an extension and adaptation of OperA [55], which has been developed for the purpose of supporting organizational interactions and collaborations in Multi-Agent Systems.

The choice of OperA is motivated as follows. Firstly, OperA proposes an expressive way for defining open organizations distinguishing explicitly between the organizational aims and the agents who act in it. Such a design choice corresponds to the requirement of this research in the sense that the specification of organizational structures, requirements and objectives is independent from any knowledge on the properties or architecture of agents, which allows participating agents to have the freedom to act according to their own capabilities and demands. Secondly, OperA copes with different organization modeling dimensions necessary for the analysis of organizational compliance. In particular, OperA's social structure defines roles and role relations, which provide an effective way of encapsulating compliance requirements, the normative structure provides a basis for the formalization of compliance rules, and the interaction structure gives a footprint for agent interaction plans. Thirdly, the OperA framework is supplemented with a comprehensive methodology for designing agent organizations in real world scenarios, which facilitates to position the application of this research.

3.3 OperA+ Architecture

OperA+ is an agent-based, multi-context, multi-level framework for designing and evaluating organizational interactions aiming at ensuring organizational compliance [114], [121], [119]. Targeting multi-organizational collaborations, OperA+ expands organizational modeling in two directions that enable analysis and decision making in different situations where interactions are performed at multiple levels towards some collective goals. One direction allows variation of details in organizational models from abstract (business values) to concrete (operational details). The other direction identifies the possible application environments (i.e., contexts) of an organizational model and elaborates the abstract specification into sets of contextual specifications according to the specific requirements of the refined contexts. The two directions together provide a contextual link between organizational values and implementation details.

3.3.1 Modeling Framework

Figure 3.1 shows the modeling framework of OperA+ which consists of four abstraction levels. Firstly, at the strategy level, the top-level objectives of an organization together with its application environment, are identified. The objectives indicate the very ends pursued by the organization. The environment represents the context in which the organization operates, and it is determined by relevant situational variables which concern but are not restricted to individuality, activity, location, time, and relation [224]. Such situational variables are mainly from two sources, i.e., external condition variables and agent feedback. *Contexts* are defined as specific subsets of the Cartesian product of the variables' value sets. They indicate organization responses (states) to various situational conditions by enabling different solutions for the achievement of organizational objectives, which decompose the top-level objectives into a set of sub domains. Specifically, a sub domain will expand if more information about the variables is obtained, shown as the area enclosed by the dotted line. Contexts are the abstraction of these sub domains and indicate different situational alternatives that may be adopted to ensure the achievement of the top-level objectives.

For each context, a set of interdependent *roles* representing different responsibilities and requiring different capabilities are identified, which is defined as a *social structure*. The social structure specifies a division of labor for a specific context in order to provide a solution of encapsulating the compliance requirements for the achievement of the top-level objectives. Given the role specifications, the solution level further specifies for each context a set of norms to indicate the compliance rules of the roles, which is referred to as a *normative structure*. The specification of a normative structure is expressed in terms of the actions of roles, and includes both constitutive norms (counts-as) and regulative norms (obligations and prohibitions), as well as the possible interrelations between norms (conjunction, choice, reparation/sanction).

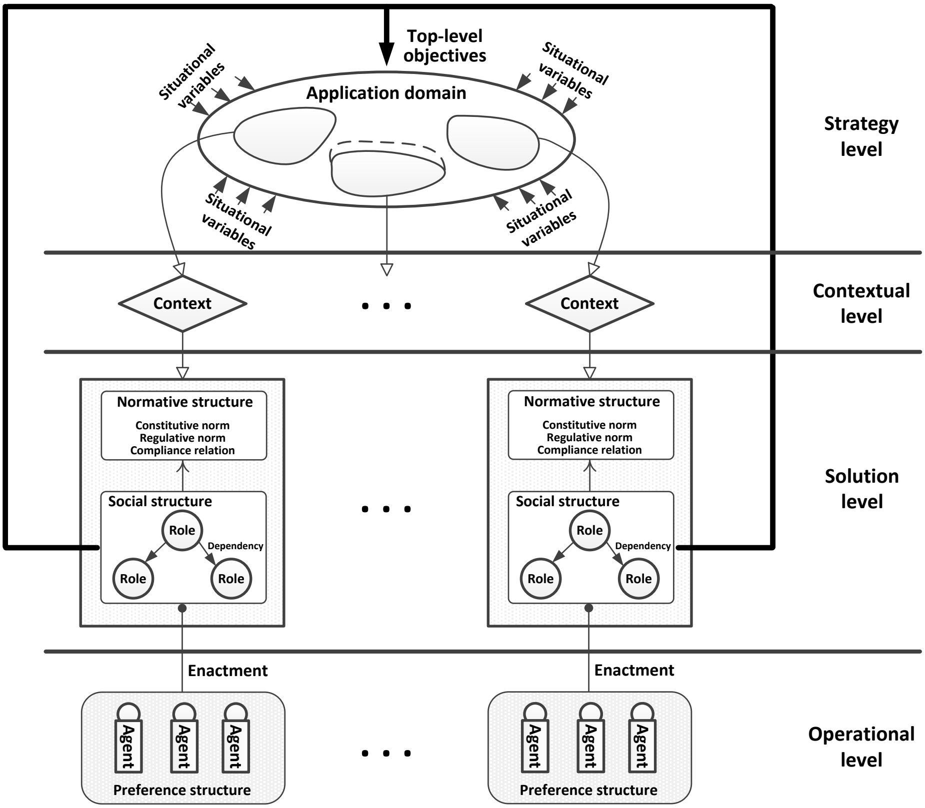


Figure 3.1: *Architecture of OperA+*

Moreover, an objective of a role at the solution level can again be decomposed into more specific components, which enables a modular way of organizational modeling. The three upper-level descriptions illustrate the compliance requirements of the organization design, which are expected to be complied with in order to achieve the overall objectives of the system.

The purpose of the organizational design by OperA+ is to specify a mechanism to coordinate the behavior of the operational *agents* that enables the organization to achieve its top-level objectives in dynamic settings. Only by an exact match between organizational roles and operational agents, such organizational behavior can be realized. Thus, at the operational level, eligible (external) agents will participate in the organization and enact the roles that are assigned to them. In order to incorporate the characteristics of the individual agents in organizational design, the preferences of the agents are captured at this level, referred to as *preference structure*. Such considerations will facilitate the performance of the organization in the sense that agent interactions are better embedded with both the compliance requirements of the organization and the autonomy desires of the individual actors.

In OperA+, organizational design can be initiated at two abstraction levels. The first is the strategy level, which specifies the top-level objectives of the organization and gradually identifies the required roles and their dependencies according to the expected functionalities of the organization from abstract to concrete. The second is the operational level that depicts the operational details of an organization from which related roles and contexts can be abstracted.

Formally, we define an organizational model designed with OperA+ as follows.

Definition 3.1 (Organizational model) *An organizational model OM is defined as a tuple (SS, NS, PS) where*

- *SS indicates a social structure,*
- *NS indicates a normative structure, and*
- *PS indicates a preference structure.*

The formalization of the social structure is described in Chapter 4, the formalization of the normative structure is described in Chapter 5 and 6, and the formalization of the preference structure is described in Chapter 7.

It has to be noticed that, in this dissertation, we overload the definitions for simplicity, e.g., OM is used both as acronym to "organizational model" as well as a reference to a specific $OM = (SS, NS, PS)$. Moreover, we use subscripts to refer to the elements of a specific model OM, e.g., SS_{OM}, NS_{OM} and PS_{OM} respectively refer to the social structure, normative structure and preference structure of the organizational model OM.

3.3.2 Modeling Reflections

OperA+ specifies a multi-agent organization in three interrelated structures, i.e., social structure, normative structure and preference structure. The design of these structures is intended to meet the four requirements that are identified in Section 3.1 for the analysis and achievement of organizational compliance. Figure 3.2 shows how different components in OperA+ contribute to those requirements.

Compliance requirements: In OperA+, the formalization of compliance requirements is enabled by the construction of social and normative structures. The social structure specifies how compliance requirements are encapsulated through the identification of roles and their dependencies. The normative structure specifies, on the basis of role specifications, the compliance rules prescribing the legitimate ways of operation described as (deontic) norms. The combination of these two structures provides a representation of organizational compliance by indicating what desired behavior (norms) should be ensured by whom (roles).

Moreover, the social structure proposes a compositional model for the specification of roles, which enables a multi-level representation of social relations

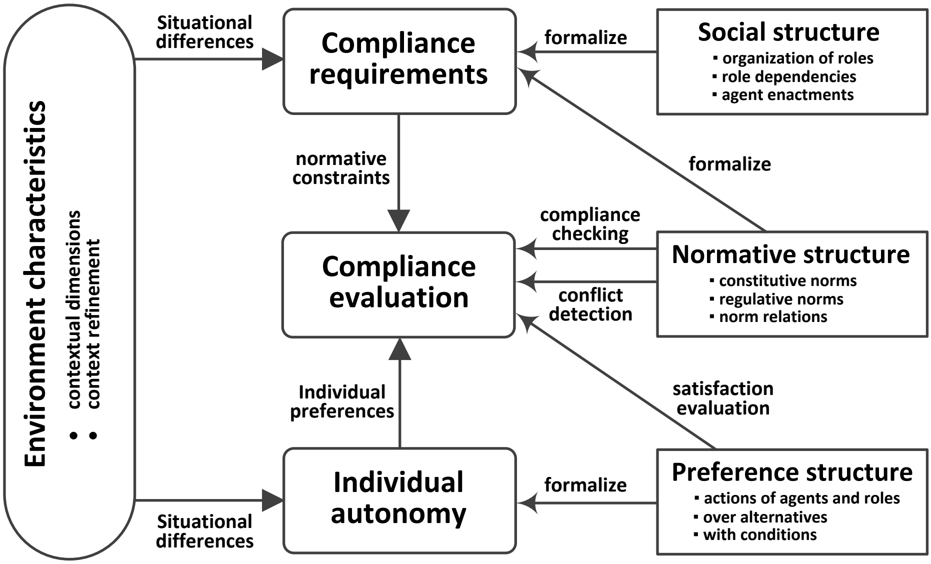


Figure 3.2: *Contribution modules in OperA+*

in organizations. As a result, OperA+ enables to build organizational models at different scales of analysis ranging from extremely broad-grained (abstract values) to exceedingly fine-grained (implementation details). The link between organizational roles and participating agents is captured by the enactment dimension of the social structure. The normative structure proposes a modular way of formalizing compliance rules by taking into account the interrelations between norms. This not only facilitates the understanding of the regulative aspects of organizational compliance, but more importantly provides an integrated view for the analysis and evaluation of organizational compliance. In addition, the normative structure differentiates between constitutive and regulative norms, in which the former indicates how participants' behavior in the physical world is perceived by an organization, and the latter describes acceptable organizational behavior through obligations and prohibitions. As a result, a link is created between what occurs in the reality and what is perceived by the organizations.

Individual Autonomy: In OperA+, the formalization of individual autonomy is enabled by the construction of preference structure, which specifies the individual preferences of the participating agents in an organization. In particular, formalisms are proposed to define the agents' preferences over their own actions, other agents' actions, and actions relating to roles. Moreover, the preference structure is capable of representing preferences over alternatives and preferences preceded by conditions. In this way, the personal interests and needs of the participants can be captured, which serves as an important source of input information for the evaluation of organizational compliance.

Environment Characteristics: In OperA+, environmental characteristics of organizational compliance are captured by the notion of context. The use of context explicitly represents the possible environments in which organizational behavior is expected. Combined with the compositional role model in the social structure, OperA+ further enables a multi-level and multi-context modeling paradigm. This not only enables designers to characterize the situations to which specific compliance requirements are applied, but also add meaningful links between different levels of descriptions. That is, refinement of high level abstractions is integrated with the refinement of environments/contexts in which more specific requirements are identified. As a result, organizational design is distributed into reusable blocks, which facilitates the establishment of organizational compliance in changing environments. Moreover, contexts also play a part in the refinement of agents' abstract preferences in changing environments.

Compliance Evaluation: In OperA+, compliance evaluation consists of two parts: (1) The first part relates to the normative structure, in which both conflict detection and compliance checking of norms are considered. Conflict detection determines whether there are any norms that cannot be complied with simultaneously. Compliance checking determines whether the participants' behavior violates any norms (obligations and prohibitions) specified in the normative structure. (2) The second part relates to the preference structure, in which we incorporate agent autonomy into the evaluation of organizational compliance by extending compliance with the analysis of preference satisfaction. Compliance evaluation is operationalized by means of Colored Petri Nets, which enables computational analysis of norm compliance and individual preferences in a unified way. As such, compliance evaluation incorporates both organizational constraints and personal needs, enabling to find a balance between conformity and autonomy.

3.4 Discussion

Given the architecture of OperA+, we now discuss the features of OperA+ in organizational modeling from the aspects of model flexibility and reusability. Flexibility relates to the ability of ensuring organizational compliance in changing environments/contexts. Reusability relates to the capability of reusing solutions of ensuring organizational compliance across different contexts. We will demonstrate these features in Chapter 4.

3.4.1 Model Flexibility

An important feature of OperA+ is that it provides mechanisms for context sensitive design of organizational interactions. That is, it uses situational variables to abstract contexts that characterize conditional alternatives of achieving organizational objectives. These contexts make changes to the organization model at a contextual level, which in turn lead to the adaptation of other structures such as social structure and normative structure. For example, we can have several refined contexts in the current design of an organization and a new context in the future design, each of which is determined by the value of relevant situational variables and refers to a sub domain in the general application domain. To this end, specific roles and role dependencies are identified for the requirements of each refined context. Such a way of structuring organizational interactions, not only adds meaningful building blocks in organizational models, but also improves model flexibility since current contexts can be easily adapted and new contexts can be easily integrated when changes occur.

3.4.2 Model Reusability

OperA+ models organizational interactions at multiple levels of abstraction. The higher levels model the systems in terms of coarser-grained components while the lower levels provide increasing details to the components designed and controlled by different entities, which facilitates modularity that can clearly describe distinctive behavioral patterns with respect to different contexts. More specifically, a top-level organization is made up of several interrelated roles which remain the same during the life cycle of the organizational design while the lower-level organizations consist of more specific entities which adapt to different situational domains. This not only enables integrating different types of components in one model and provides necessary opacity since designers can decide how "deep" a model is constructed, but also ensures that components and groups of components can be easily reused at all levels of design.

3.5 Conclusions

In this chapter, we presented an overview of the agent-based framework OperA+ for modeling and evaluating organizational compliance. OperA+ not only provides constructs to capture compliance requirements from a system point of view but also takes into account the internal preferences of individual agents in the evaluation of organizational compliance.

The OperA+ framework is based on (1) research in organization modeling which provides the insights on how compliance knowledge is formed and expressed in the creation of organizational structures, and (2) research in Artificial Intelligence which provides the methods on how such knowledge can be formalized and evaluated with automated support. By adopting the MAS paradigm, the problem of achieving organizational compliance is well captured by the regulation and coordination problem of agent interactions. Moreover, by combining contextualization with compositional modeling, OperA+ provides a flexible representation of how organizational compliance is coupled with environmental characteristics in a refinement process, which will facilitate the understanding and communication of compliance requirements in organizational collaborations.

The contributions of OperA+ to organization modeling are threefold. First, it proposes a multi-level multi-context framework for modeling and analyzing agent interactions. Multi-level modeling itself is not new, e.g., the multi-level agent interaction protocol in [162], the multi-level normative model in [208], the multi-level organizational model in [27]. However, integrating contextual differences and refinement at multiple levels has not been explicitly considered by the existing agent organization frameworks. Second, it provides an integrated representation of the behavioral constraints on agent organizations based on the notion of norms. The use of norms in MAS has been studied by many researchers, but the inter-relations between norms are not investigated in a comprehensive way. Third, it introduces a mechanism to incorporate the characteristics of individual agents in organizational design, which enables the evaluation of organizational interactions on the basis of both organizational requirements and individual needs.

Chapter 4

Social Structure

This chapter has the following contributions:

- a multi-level role model for specifying social relations and agent participation in organizations, and
- a mechanism for contextualizing specifications of social relations

The construction of the social structure of an organizational model is to provide a way of encapsulating compliance requirements of the organization, which indicates the responsibilities of the participants who are expected to collectively achieve the organizational objectives. For this purpose, we adopt the notion of role which rests on a theatrical analogy [82]: individuals in a society occupy positions, and their role performance in these positions is determined by demands and rules. The role perspective assumes that individuals are members of social positions and hold expectations for their own behavior and those of others, and performance results from the social prescriptions and behavior of others [22]. Based on these ideas, the social structure of OperA+ encapsulates the compliance requirements of an organization by specifying the roles holding in the organization, including their objectives, requirements, groups of roles, and the relations between roles, which are independent of the individuals that will eventually occupy the roles in the organization. These roles indicate the responsibilities assigned to, and the relationships existing among, the participants in the organization for the achievement of organizational objectives.

Parts of this chapter have been published. Section 4.1 and 4.2 are based on our contribution to the 13th International Workshop on Coordination, Organisations, Institutions and Norms @ IAT 2011, pp. 58-74, 2011 [119] and the International IFIP Electronic Government Conference, pp. 308-319, 2011 [120].

In practice, roles in organizations are usually structured into work groups, departments, divisions, and the organizations in turn may be seen as roles within larger organizations. Such organizational structures reflect important requirements impressed on them by the context to which the organization must adapt. This suggests that to coordinate and regulate the behavior of an organization, it may be as important to look outside the organization and its context as to look inside the organization at its component units. In this sense, the establishment of compliance requirements needs to cover multiple levels of the participants' activities and interactions, and reflect upon the specific contexts where the activities and interaction occur.

To this end, we present a multi-level role model for the specification of social relations in organizations and show how contextual information is integrated to structure such specifications. In a nutshell, an organization itself can be seen as a role (or a set roles), whose ultimate objectives are usually somewhat vague and abstract. Such objectives can serve as the starting point for the construction of means-ends chains that involve: (1) starting with the general objectives to be achieved which can be captured by a set of very generally specified roles, (2) discovering a set of divisions of the top-level roles with respect to various contexts for accomplishing the objectives of those role in differing situations, and (3) taking each of the sub-roles in the divisions of the top-level roles, recursively, as new sub-objectives and discovering a set of more detailed divisions of the sub-role for achieving the associated sub-objectives with respect to a set of refined contexts. Moreover, based on the specification of organizational roles, we investigated the enactment dimension of social structures which provides a link between the roles and the participants in an organization, i.e., enactments of the roles by organizational participants. In this way, the compliance responsibilities of organizational participants can be encapsulated into modular and meaningful blocks that reflect the changes of requirements of organizational compliance with respect to the changing contexts where the organization is situated.

This chapter is organized as follows. Section 4.1 introduces the fundamental concepts employed in the social structure of an OperA+ model. Section 4.2 gives the formalization of a specification and enactment with respect to a social structure. Finally, we present the conclusions in Section 4.3.

4.1 Fundamental Concepts

In this section, we present all the fundamental concepts that are needed for the construction of the social structure and interaction structure of OperA+ models in this chapter.

4.1.1 Role

In organizations, roles are assumed to be associated with identified social positions and to be generated by normative expectations [21]. They present the standards for the recruitment of participants in an organization and the link to the evaluation of the participants' performance. In this sense, the specification of roles in an organization can be an effective way to encapsulate compliance requirements for the achievement of organizational objectives. Furthermore, a role in an organization may be further elaborated into finer-grained components that can collaboratively accomplish the objectives of the role, enabling a flexible way of modeling social relations. Considering all these aspects, we give the definition of roles as follows.

Definition 4.1 (Atomic role) *An atomic role* r_A *is defined as a tuple* $(name, Obj, Rcap)$ *where*

- *name is the identifier of* r_A,
- *Obj is a set of objectives assigned to* r_A,
- *Rcap is a set of capabilities required by* r_A *to accomplish Obj,*

Definition 4.2 (Composite role) *A composite role* r_C *is defined as a tuple* $(name, Obj, Rcap, org)$ *where*

- *name is the identifier of* r_C,
- *Obj is a set of objectives assigned to* r_C,
- *Rcap is a set of capabilities required by* r_C *to accomplish Obj,*
- $org = (R, Dep)$ *is an organization where*
  - *R is a set of atomic or composite roles such that* $Obj_{r_c} \subseteq \bigcup_{r\in R} Obj_r$ *and* $Rcap_{r_c} \subseteq \bigcup_{r\in R} Rcap_r$, *and*
  - *Dep is a set of role dependencies such that an element from Dep is defined as* $dep = (r, r', obj)$, $r, r' \in R$, $obj \in Obj_r$.

A set of objectives *Obj* indicates the desired results an organization envisions, which are encapsulated in the specification of roles. To ensure the achievement of the corresponding objectives, a set of capabilities *Rcap* are required for the actors who seek to occupy a specific role. Capabilities represent skills or capacities that can be acquired by actors such that they have the power or ability to carry out certain tasks. Following the approach in OperA [55], both objectives and capabilities are expressed as predicates in first order logic.

The difference between atomic roles and composite roles is that composite roles have an additional component *org* defined as a set of interdependent roles.

This indicates that a composite role is further elaborated as an organization of sub-roles. In this case, the union of the objectives of the roles in R should cover that of the composite role r_C, and the union of the required capabilities of the roles in R also needs to cover that of the composite role r_C.

Following the definition presented in [55], the dependency relation between two roles r and r' for an objective obj, represented as (r, r', obj), indicates that r depends on r' for the realization of obj. Such relations between roles describe how actors can interact and contribute to the realization of the objectives of each other. That is, an objective of a role can be delegated to or requested from other roles. The normative aspect of roles will be presented in the next chapter.

For example, we can have an atomic role *PhD_student* with a set of objectives such as getting good publications, getting good academic network, writing a good dissertation, etc. Moreover, the role might require capabilities such as good writing skills, good communication skills, independent research ability, etc. As for composite roles, an example could be a role *Knowledge_disseminator* in a multi-disciplinary research project, and this role has an inner structure of two sub-roles *ICT_disseminator* and *Logistic_disseminator* [119]. The sub-role *ICT_disseminator* aims at disseminating knowledge among commercial service providers and the sub-role *Logistic_disseminator* aims at disseminating knowledge among logistics service providers. As for role dependency, for example, in a conference scenario the enactor of role *PC-Chair* can delegate the objective paper-reviewed to an enactor of role *PC-member*.

It can be seen from the definition of a role that there are two types of role relations. The first type is a *dependency relation* by which the accomplishment of the objectives of a role can be delegated to or requested from another role. The second type is a *decomposition relation* by which the objectives of a composite role can be disassembled into an organization of subunits. The dependency relation is between two roles at the same level and they are complementary parts of a specific organizational design. The decomposition relation is between two roles at different levels and the role at the lower level is a part of a specific refinement of the role at the higher level.

Roles specify the possible decomposition of organizational objectives. Only by putting participants into the roles can such organizational objectives be realized. Now we present the concept of agent that is used in OperA+ to model organizational participants.

4.1.2 Agent

Organizational participants are individual entities that are capable of acting autonomously [186, 192]. They can be a human being, a piece of software, or an organization of individual actors, and they possess various capabilities that enable them to reach specific goals. To formalize the participants in organizations, we adopt the notion of agent from agent theory [217], based on the analogy that an agent is an encapsulated computer system that is capable of flexible, autonomous action in some situated environment in order to meet its design objectives.

Corresponding to the required capabilities of roles, an agent possesses a set of capabilities that allow the agent to carry out specific roles and achieve the assigned objectives. Since the capabilities of individual agents may change over time, the ability of the agents to play specific roles also changes. Moreover, an agent can be an atomic entity whose inner structure is either invisible or unimportant to the other parts of the system. It can be an individual, a service, or even a company whose inner structure is hidden from the outside. An agent can also be a composite entity whose inner structure is visible and important to the other parts of the system. For example, a rescue team can be seen as a composite agent, which includes a number of individuals with various capabilities. Considering all these aspects, the formalization of agents is given as below.

Definition 4.3 (Atomic agent) *An atomic agent ag_A is defined as a tuple* $(name, Acap)$ *such that*

- *name is the identifier of ag_A,*
- *Acap is a set of capabilities possessed by ag_A,*

Definition 4.4 (Composite agent) *A composite agent ag_C is defined as a tuple* $(name, Acap, Ag)$ *such that*

- *name is the identifier of ag_C,*
- *Acap is a set of capabilities possessed by ag_C,*
- *Ag is a set of atomic or composite agents.*

The difference between atomic agents and composite agents is that composite agents have an additional component Ag defined as a set of agents. This indicates that a composite agent is composed of a number of sub-agents. As an example, at a specific level of abstraction every human being can be seen as an atomic agent that possesses different capabilities. As for composite agents, an example could be a Smith_family consisting of two individuals *Alice* and *Bob*. Moreover, organizations, such as a university, can also be modeled as a composite agent consisting of a number of sub-agents. Note that the definition of composite agent focuses on the integrity aspect of a group of entities. That is, when we consider a group of agents as a composite agent, the members in the group cannot be replaced by other agents even when they have the same capabilities.

Through the enactment of roles, agents contribute to the achievement of organizational objectives. Organizations, however, are not self-sufficient but all depend on the relations they establish with the environment of which they are a part. In the next subsection, we introduce the notion of context for the representation of the environments where organizations exist.

4.1.3 Context

No organization is self-sufficient, but exists in a specific physical, technological, cultural, and social environment to which it must adapt [186]. Organizational structure reflects important features borrowed from or impressed on it by the environment. To formalize the environment of organizations, we adopt the notion of context which refers to any information that can be used to characterize the situation of an entity [52]. In particular, we borrow the idea from the context model proposed in [80] for the representation of context. The model is based on three elements: a set of parameters, a range of values for each parameter, and a state of affairs or a domain, which draws a sort of boundary between what is in and what is out. Correspondingly, we characterize a context by a set of contextual dimensions, each of which has a structured set of values. In this way, a context provides a combination of different perspectives on the observation of the environment where organizations exist. Definition 4.5 gives the formalization of contextual dimensions.

Definition 4.5 (Contextual dimension) *D is a set of contextual dimensions where $V \in D$ is a set of values for a situational variable.*

Contextual dimensions concern but are not restricted to aspects such as individuality, time, location, activity, and relations [224]. The values of each contextual dimension are assumed to come from some structured domain. For example, we can have a contextual dimension of $D = \{Location\}$ with a value set $V = \{Netherlands,\ France, Germany, Italy, ...\}$.

With contextual dimensions, a context is then modeled as a specific subset of the Cartesian product of the contextual dimensions' value sets, formalized in Definition 4.6.

Definition 4.6 (Context) *A context $ctx \subseteq \prod_i V_i$ where $V_i \in D$ characterizes a set of situations with respect to a set of contextual dimensions.*

Contexts are indicated by different combination of contextual dimensions and the values those contextual dimensions can take. For example, we can build a context based on two contextual dimensions *Location* and *Time* and the values the two dimensions can take are respectively $\{Netherlands,\ Belgium,\ Luxembourg\}$ and $\{2013,\ 2014\}$. Two contexts can refer to the same set of contextual dimensions and take different value sets of those dimensions. For example, we can have a context representing park and another context representing office which can be characterized by a contextual dimension of location.

Given the concepts presented above, we now illustrate the construction of social structures.

4.2 Social Structure

Following the description in [119], the social structure of an OperA+ model comprises of two dimensions: (1) **specification dimension** indicates the possible decompositions of organizational objectives through a set of roles, and (2) **enactment dimension** indicates the possible populations of agents that can realize the objectives by taking up the roles.

4.2.1 Specification Dimension

The specification dimension of a social structure identifies the objectives to be achieved and the division of labor to achieve the objectives, i.e., what roles are needed and how they interrelate with each other. The specification dimension starts from a set of interrelated roles encapsulating the overall objectives of an organization, each of which can be further elaborated by an organization of sub-roles. An organization can be seen as a social arrangement which pursues collective objectives through a set of connected roles [55]. Roles are typically declarative concepts meant to represent a part of the organization's design. OperA+ includes two kinds of roles: atomic role and composite role. Each composite role refers to an organization of sub-roles at a lower level in the hierarchy which elaborates the objectives of the composite role into finer-grained components and gives more constraints or information on the how the objectives can be achieved. Atomic roles are not further specified enabling heterogeneous enactment. Definition 4.7 gives the formalization of a specification of a social structure.

Definition 4.7 (Specification of social structure) *A specification of a social structure* SS^s *is a defined as a tuple* (R, Dep) *where*

- R *is a set of roles, and*
- Dep *is a set of role dependencies.*

The specification may consist of a single role and thus role dependencies Dep can be an empty set. A role, as shown in Definition 4.2, is a tree-like structure consisting of sub-roles and role dependencies at different abstraction levels. That is, a specification of a social structure is a set of hierarchically organized components and their interacting relations.

To simplify the following definitions based on the specification of a social structure, we define the function *unbox*. Given a role r, *unbox* returns all the sub-roles derived from r.

Definition 4.8 (Unbox) *Given a role* r*, the function unbox is defined as follows:*
– *if* r *is an atomic role:* $unbox(r) = \{r\}$
– *else:* $unbox(r) = R_{org_r} \bigcup (\bigcup_{r' \in R_{org_r}} unbox(r'))$.

By the function *unbox*, the set of all sub-roles in a given role can be obtained. As a sub-role is either atomic or composite, the set of all sub-roles is divided into two subsets, i.e., the set of all atomic roles and the set of all composite roles. Given a role r, we can obtain the two subsets respectively by the following functions.

- $R_A(r) = \{r' | r' \in unbox(r) \wedge r' \textit{ is an atomic role}\}$,
- $R_C(r) = \{r' | r' \in unbox(r) \wedge r' \textit{ is a composite role}\}$.

A specification of a social structure presents how organizational objectives may be decomposed by means of interrelated roles. To accomplish the organizational objectives, a description from an implementation perspective is needed, i.e., the possible populations of agents that can take up the roles to collaboratively pursue the objectives. To this end, we now illustrate the construction of the enactment dimension of a social structure.

4.2.2 Enactment Dimension

Given a specification of a social structure, the enactment dimension provides an illustration of the possible agent populations that can occupy the specified roles such that the corresponding objectives can be realized. Since our model targets open systems in which agents are not known at design time, the enactment dimension is only an illustration of the possible populations of agents that can match the requirements of the specification dimension. In OperA+, an enactment of a social structure, with respect to a specification of the social structure, is defined as a set of agents and their mappings to the roles defined in the specification, as formalized in Definition 4.9.

Definition 4.9 (Enactment of social structure) *The enactment of a social structure SS^e, with respect to the specification of the social structure SS^s, is defined as a tuple (SS^s, Ag, RE) such that*

- *SS^s is a specification of a social structure,*
- *Ag is a set of agents,*
- *RE is a set of role enactments where $re = (r, ag)$, $r \in \bigcup_{r' \in R_{SS^s}} unbox(r')$, $ag \in Ag$,*
  - $\forall (r, ag) \in RE : Rcap_r \subseteq Acap_{ag}$,
  - $\forall (r, ag) \in RE$ *and r is a composite role:*
    ag is a composite agent and $\forall r' \in R_{org_r}, \exists ag' \in Ag_{ag} : (r', ag') \in RE$.

Ag specifies all the agents participating in an enactment. RE specifies a set of role enactment relations that indicate which agents enact which roles. A requirement for an agent to enact a role is that the possessed capabilities of the agent should meet the required capabilities of the role, indicated by the condition $\forall (r, ag) \in RE : RCap_r \subseteq Cap_{ag}$. Moreover, in a role enactment relation (r, ag), if the role r is a composite role, the agent ag should contain corresponding sub-agents for the enactment of the sub-roles, indicated by the last condition $\forall (r, ag) \in RE$ *and* r *is a composite role*: $\forall r' \in org_r, \exists ag' \in Ag_{ag} : (r', ag') \in RE$.

It can be seen that Definition 4.9 only describes the formation rules of an enactment but does not specify whether a specification is fulfilled by an enactment. Therefore, we give the definition of complete enactment as follows.

Definition 4.10 (Complete enactment) *An enactment of a social structure* $SS^e = (SS^s$, Ag, $RE)$ *is a complete enactment if* $\forall\ r \in \bigcup_{r' \in R_{SS^s}} R_A(r')$, $\exists\ ag \in Ag : (r, ag) \in RE$, *indicating that for each atomic role in the specification* SS^s, *there is at least one agent enacting it.*

The complete enactment indicates that every atomic role specified in the social structure is enacted by an agent. In this way, the composite roles are indirectly enacted by the agents enacting the sub-roles.

4.2.3 Combining the Two Dimensions

With the formalization of specification and enactment presented above, the definition of a social structure is summarized as follows.

Definition 4.11 (Social structure) *A social structure* SS *is defined as a tuple* $(SS^s$, $SS^e)$ *such that*

- SS^s *is a specification of the social structure* SS, *and*
- SS^e *is an enactment of the social structure* SS *with respect to* SS^s.

Figure 4.1, depicted in UML class diagram, shows the core elements and the relationships between them consisting in a social structure. From the perspective of roles, an atomic role can be enacted by any type of agents while a composite role can be (1) enacted by a composite agent providing that the internal organization of the agent matches that of the composite role, or (2) enacted by a set of independent agents each enacting a sub-role respectively. From the perspective of agents, each agent can enact one or more roles if its capabilities meet the requirements of the roles. A specification of social structure can have multiple enactments. For one enactment, there are a set of agents enacting the roles in the specification. Some of the agents may have their own understanding of the objectives of the roles they enact. Therefore, agents can extend the inherited specification according to their own capabilities, i.e., they may further refine the specification to better achieve the objectives.

With a set of hierarchically organized roles, a social structure provides an evolutionary understanding on how the objectives of an organization are decomposed.

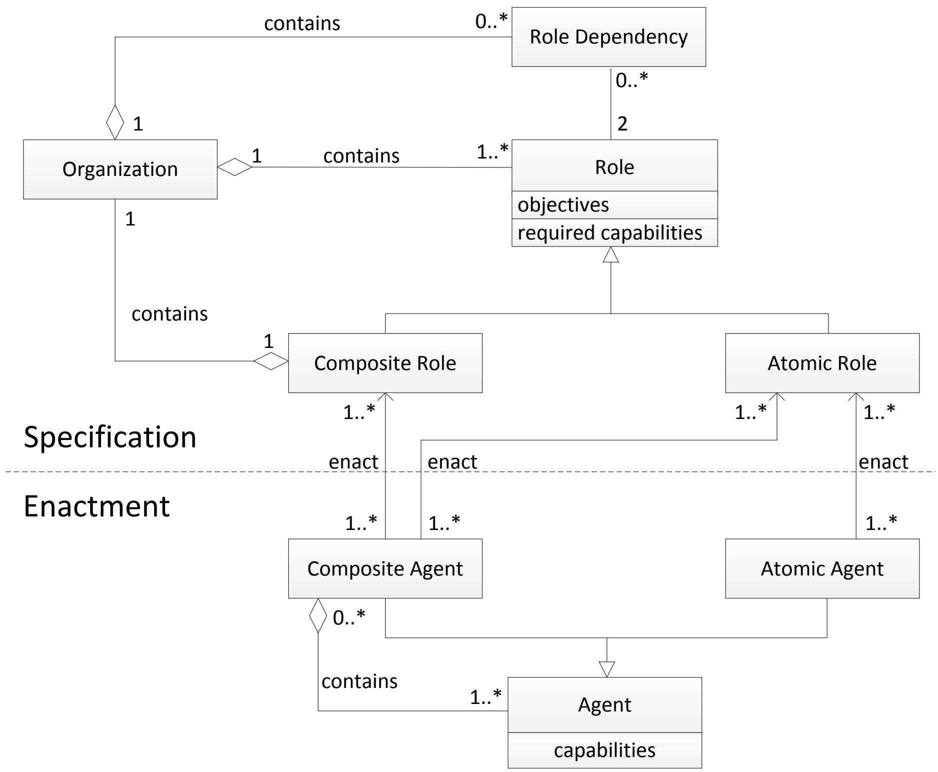


Figure 4.1: *Meta-model of the social structure of OperA+*

However, the question still remains on how such constructions are established, i.e., what is the driving force behind this design pattern. In the next section, we investigate this issue.

4.2.4 Contextualization

In practice, roles in organizations are usually structured into work groups, departments, divisions, and organizations in turn may be seen as roles within larger organizations. Such organizational structures reflect important requirements impressed on them by the context to which the organization must adapt. This is to suggest that contexts play an important part in the structuring of organizational roles. To this end, we integrate the context model presented in Section 4.1 in the specification of social structures to indicate how context information can impact the decomposition of organizational objectives by means of roles.

Figure 4.2 shows how contexts are linked to the specification and the enact-

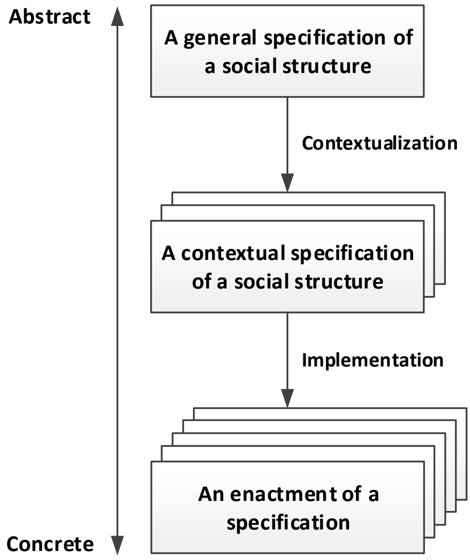


Figure 4.2: *Modeling levels of social structure*

ment of social structures. Firstly, a general specification of a social structure is constructed to represent the ultimate objectives served by an organization in an abstract way. Such a specification is intentionally specified at a high level of abstraction to range over many different situations and to require little maintenance over time. Definition 4.12 gives the formalization of a general specification.

Definition 4.12 (General specification of social structure) *A specification SS^s of a social structure is a general specification, denoted as SS^{gs}, iff $\forall r \in R_{SS^s} : r$ is an atomic role.*

A general specification of a social structure contains a set of interdependent roles with no inner structure, i.e., the roles are atomic. In this way, the general specification gives a relatively abstract description of the responsibility decomposition for the achievement of organizational objectives.

With respect to a general specification of a social structure, various environmental conditions may exist and pose different requirements for the fulfillment of the atomic roles presented in the general specification. In this sense, the general specification needs to be refined into a set of contextualized specifications that provide context-specific decompositions of the general objectives carried by the roles in the general specification. Practically, by applying contextual information to a general specification, some of the atomic roles are further elaborated and transformed into composite roles such that more information or constraints are added on which sub-roles are needed and how they are related, as captured by a contextualization of social structure shown in Definition 4.13.

Definition 4.13 (Specification contextualization of social structure) *A specification contextualization of a social structure is defined as a tuple* $(ctx, SS^{gs}, SS^{cs}, RM)$ *where*

- ctx *is a context as described in Definition 4.6,*
- SS^{gs} *is a general specification of social structure,*
- SS^{cs} *is a specification of social structure according to Definition 4.7,*
- $RM \subseteq R_{SS^{gs}} \times R_{SS^{cs}}$ *such that*
  - $\forall r \in R_{SS^{gs}}, \exists!(r, r') \in RM : r' \in R_{SS^{cs}}$ *and* $name_r = name_{r'}$ *and* $Obj_r \subseteq Obj_{r'}$ *and* $Rcap_r \subseteq Rcap_{r'}$
  - $\exists(r, r') \in RM : r'$ *is a composite role*

In the specification contextualization, a set of role mappings RM is defined between the set of roles in a general specification SS^{gs} and a specification SS^{cs} (we call SS^{cs} a contextual specification of social structure). Firstly, for each role in the general specification, there exists another role in the contextual specification such that (1) the two roles have the same name, (2) the set of objectives of the role from the contextual specification covers that of the role from the general specification, and (3) the set of required capabilities of the role from the contextual specification covers that of the role from the general specification. These imply that the set of roles in the general specification are inherited by the contextual specification. Moreover, there exists at least one role mapping between a role in the general specification and another role in the contextual specification such that the role from the contextual specification is a composite role. This is to indicate that some of the roles in the general specification are further elaborated into organizations of sub-roles by the contextual specification. Therefore, the contextual specification can be seen as an extension of the general specification with respect to a specific context, by means of elaborating some of the atomic roles in the general specification into composite roles.

Figure 4.3 gives an example of a graphical illustration on how a general specification of social structure may be contextualized into a set of contextual specifications with respect to various contexts. Different contexts may have different requirements on different sets of roles in the general specification, and thus a role can be extended to different lower-level organizations of interrelated sub-roles. In this way, the general objectives pursued by an organization are factorized into context-specific sub-objectives that can be assigned to organizational subunits. From a general specification to a set of contextual specifications, the factorization of general objectives into specific sub-objectives that can be assigned to organizational subunits under different environmental conditions enhances the regulation of agents' behavior from an organizational perspective.

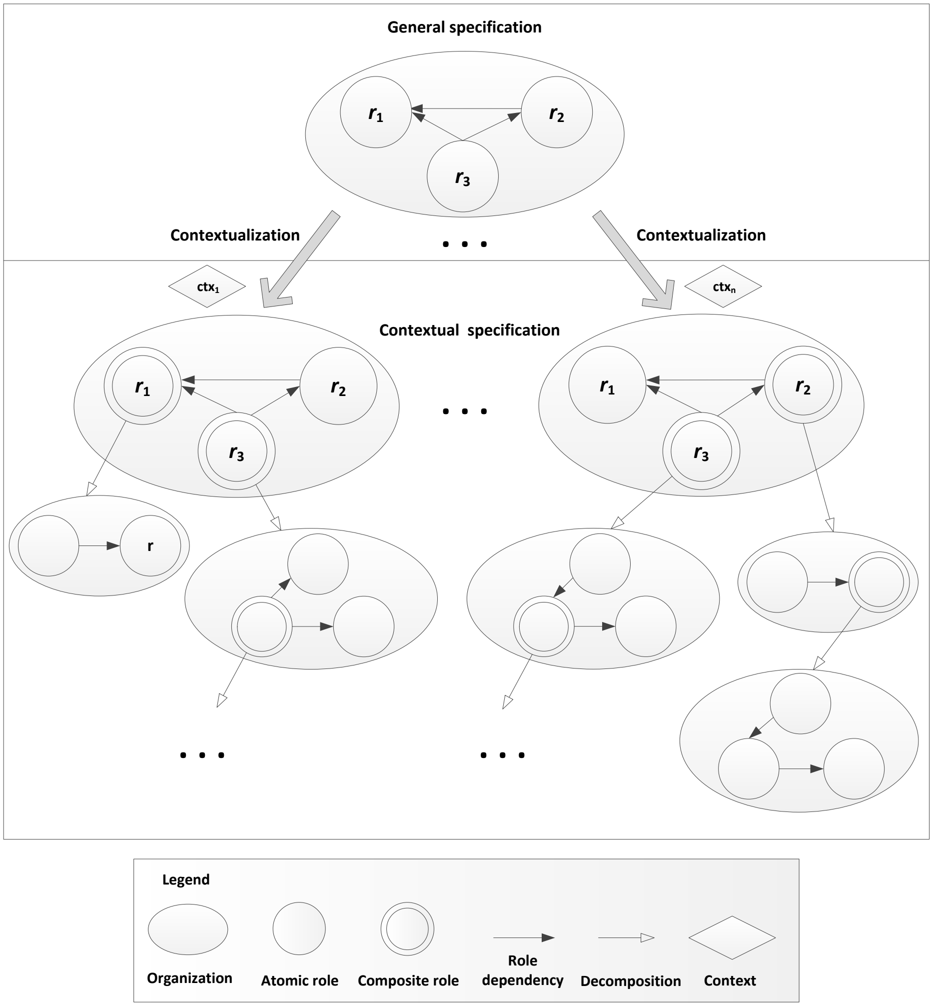


Figure 4.3: *Contextualization of social structure*

The contextual specifications of a social structure present how organizational objectives are decomposed by means of elaborating the inner structure of abstract roles with respect to differing contexts. Finally, in order to realize the organizational objectives, an implementation of those specifications is needed, i.e., participants need to be employed to enact the roles. For each contextual specification, there can be multiple enactments, i.e., a mapping from different populations of agents to the roles in the specification. This can be reflected from the fact in practice that the employees of an organization may change from time to time.

4.3 Conclusions

In this chapter, we presented the formalisms for the construction of the social structure of OperA+. Our goal was to provide a flexible and expressive way of encapsulating the compliance requirements assigned to and the relationships existing among the participants in an organization. In line with the essence of the OperA+ framework discussed in the previous chapter, the construction of a social structure follows a hierarchy in terms of role decomposition embedded with contextualization, which promotes the consistency of decisions and activities from abstract values to implementation details.

The two dimensions enclosed by a social structure, on the one hand, present the regulative aspects of an organization in terms of objective decompositions, and on the other hand, provide the possible agent populations that can collaboratively accomplish the ultimate objectives served by the organization. The concept of composite roles and agents provide designers with a flexible way of organizing the responsibilities and opportunities envisioned by an organization. The higher-level specification captures the commonalities of organizational collaborations while the lower-level specifications present the individualities by layers upon layers of elaboration according to more specific requirements. Components at the same level are modeled separately through lower-level specifications, which decreases their mutual influence when one of them changes.

Additionally, our model of social structure enables the designers to determine a balance between conformity and autonomy. That is, one can ensure conformity by articulating more finer-grained components (roles and role relations), while, on the other hand, provide more autonomy by leaving the specification open and abstract. The lower the social structure extends, the more constraints the organization imposes on the behavior of the participating agents and thus limit the range of decisions that the participating agents can make. Therefore, whether an organization gives more conformity or autonomy is determined by the extent to which the organization is formalized.

Chapter 5

Normative Structure:
Definition and Compliance Checking

This chapter has the following contributions:

- a normative language for specifying norms and their compliance relations,
- a computational mechanism for checking norm compliance with respect to agent interactions.

In the previous chapter, we have shown the social structure of OperA+ which provides a way of encapsulating compliance requirements by specifying the roles of an organization and the enacting relations between the roles and the agents participating in the organization. In this chapter, we present the normative structure of OperA+ which introduces a prescriptive and evaluative dimension into an organizational model to capture the compliance rules that are used by organizations to regulate the behavior of participants on how things should be done, i.e., the legitimate means to pursue the objectives of the organization. Some rules are applicable to all members of an organization while others apply only to selected types of actors. Thus roles share in the regulating character of an organization in the sense that compliance rules indicate the relationships among roles in terms of what are incumbent of one role owes to incumbents of other roles. As soon as actors are considered as role enactors, their behavior is ipso facto susceptible to enforcement [19].

From an organizational perspective, participants are expected to follow the compliance rules that are relevant to the roles they enact in the organization.

This chapter is based on our contributions to the Journal of ACM Transactions on Management Information Systems, to be published, [115] and to the 12th International Conference on Autonomous Agents and MultiAgent Systems, pp. 1121–1122, 2013 [118].

However, participants are autonomous entities who have the capability of controlling their own actions. Therefore it is necessary for the organization to check whether the actual behavior of the participants is in accordance with the compliance rules. This requires both a formalism of capturing compliance rules and a mechanism of evaluating the participants' behavior. For these purposes, we propose a normative language called Norm Nets (NNs) to provide a formalization of compliance rules, and a mapping from NNs to Colored Petri Nets (CPNs) to enable the verification of participants' behavior.

Compliance rules are applied to the participants through the roles they enact in the organization and may come from various regulation sources such as legal regulations, business contracts, internal policies, etc. They can be fairly complex in terms of the conditions, targets and scopes they refer to. The possibility of interrelations between them brings added complexity. Moreover, it is recognized that there is a distinction between regulative rules and constitutive rules [188], in which regulative rules concern what ought to be the case by regulating antecedently existing activities while constitutive rules do not merely regulate but also create the very possibility of certain activities. All these aspects of compliance rules are considered in the formalism of NNs as well as the mechanism of checking compliance of participants' behavior.

This chapter is organized as follows. Section 5.1 introduces the normative language Norm Nets (NNs), including the components needed for norm specification and the state transitions of norms. Section 5.2 presents the operational semantics of NNs through a mapping to CPNs. Section 5.3 gives an analysis of the properties of NNs in terms of the types of norms and normative relations that can be represented by NNs. Section 5.4 illustrates the mechanism of checking norm compliance of agent interactions. In Section 5.5, we discuss the characteristics and limitations of NNs by comparing it with other approaches. Section 5.6 concludes this chapter.

5.1 Norm Nets

The construction of the normative structure of an organizational model is to capture all the compliance rules that are relevant to the regulation of the organizational participants. Compliance rules, e.g., legal regulations, usually describe what should (not) be done rather than how something can be done. In normative systems [154, 13], the notion of norm has been proposed to specify the (un)desired behavior of agents by means of the deontic concepts: obligation, prohibition and permission. Norms provides a precise and unambiguous specification of the normative constraints the entities are subject to in the context where the normative system applies. Both compliance rules and norms have a prescriptive nature and applying normative theories to the modeling of compliance rules has been explored by many researchers (e.g., [90], [47]).

Moreover, as we have illustrated earlier, compliance rules are not independent but are interrelated in different ways, and there is a distinction between regulative and constitutive rules. In order to formalize all these aspect of compliance rules, we now present the normative language Norm Nets (NNs) [115] which is based on the notion of norm from normative systems, and show how it can be used to formalize compliance rules and facilitate evaluation of participants' behavior in organizations. In the remainder of this section, we refrain from presenting underlying semantics for the different elements of NNs. The semantics will be introduced later in Section 5.2. An intuitive reading of the elements suffices for now.

5.1.1 Preliminaries

In normative specifications, we differentiate between two types of events: (1) *external events* are a reflection of who does what in the physical world by specifying the agents and the actions they perform, and (2) *institutional events* represent who does what from the perspective of the organization by specifying the roles that are assigned to the agents and the actions that are available to the role-enacting agents. Accordingly, we use agent-action pairs to capture the occurrence of external events in the physical world and use role-action pairs to represent institutional events perceived by the organizations.

Definition 5.1 (Event) *Let R be a finite set of roles, Ag be a finite set of agents, and $\mathcal{A}$ be a finite set of actions. The set of events E is defined by the union of two disjoint sets $Ev \cup Iv$ where*

- $Ev = \{e_1, \ldots, e_m\}$ *is the set of external events such that* $e_i = (ag, \alpha), 1 \leq i \leq m, ag \in Ag, \alpha \in \mathcal{A}$,
- $Iv = \{\varepsilon_1, \ldots, \varepsilon_n\}$ *is the set of institutional events such that* $\varepsilon_j = (r, \alpha), 1 \leq j \leq n, r \in R, \alpha \in \mathcal{A}$.

An *external event* $e = (ag, \alpha)$ describes the fact that an agent ag performs an action α, indicating the occurrence of an event in the physical world. For example, we can express an observation "Bob enters a building" by defining an external event (*Bob*, *enter_building*). An *institutional event* $\varepsilon = (r, \alpha)$ describes an action α of a role r. For example, we can express an institutional observation "a student enters the library" by defining an institutional event (*Student*, *enter_library*). As a convention, the name of agents and roles will start with a upper-case letter while the name of actions will start with a lower-case letter. Using the notion of event, next we introduce a propositional language L_E over the set of events.

Definition 5.2 (Event language) *Given an event $\epsilon \in E$, let the event language L_E be the set of expressions generated by the following grammar:*

$$\varphi ::= \epsilon | (\varphi \wedge \varphi) | (\varphi \vee \varphi) | (\varphi < \varphi) | \lambda$$

$\varphi_1 \wedge \varphi_2$ *indicates both* φ_1 *and* φ_2 *occur (conjunction),* $\varphi_1 \vee \varphi_2$ *indicates either*

φ_1 or φ_2 occurs (disjunction), $\varphi_1 < \varphi_2$ indicates φ_1 occurs before φ_2 (sequence), and λ represents an empty event. We define L_e as the external event language derived from L_E with only $\epsilon \in Ev$, and L_ε as the institutional event language derived from L_E with only $\epsilon \in Iv$. We use E_φ to indicate all the events contained in φ. When φ is not a single event, we call it an event formula, and use LT_φ to indicate the left-hand side component of φ, RT_φ to indicate the right-hand side component of φ, CO_φ to indicate the combining operator of the two components.

For example, if $\varphi_1 = (Bob, enter\_building)$ and $\varphi_2 = (Librarian, give\_permission)$ where *Bob* is an agent and *Librarian* is a role, then $\varphi_1 \wedge \varphi_2$ means that Bob enters a building and Librarian (agents enacting the role of librarian) gives a permission; $\varphi_1 \vee \varphi_2$ means that either Bob enters a building or Librarian gives a permission; $\varphi_1 < \varphi_2$ means that Bob enters a building before Librarian gives a permission.

With the event language L_E, now we introduce the formalisms of NNs.

5.1.2 Regulative norms

Regulative norms prescribe how agents ideally should or should not behave. Deontic concepts like obligations, prohibitions and permissions are widely used to model such prescriptions, providing a precise and unambiguous specification [154]. In NNs, two types of regulative norms are defined, i.e., obligations and prohibitions. We leave out permissions because our focus is on the compliance status of organizations while it is not possible to violate a permission.

In her study of institutional diversity, Ostrom [165] proposed five components for the representation of institutional statements: Attribute, Deontic, AIM, Conditions, and OR else, being referred to as ADICO. Attribute is a holder for any value of a participant-level variable that distinguishes to whom a rule applies; Deontic is a holder for the three modal verbs using deontic logic: *may* (permitted), *must* (obliged), and *must not* (forbidden); AIM is a holder that describes particular actions or outcomes to which the deontic is assigned; Conditions is a holder for those variables which define when, where, how, and to what extent an AIM is permitted, obligatory, or forbidden; Or else is a holder for those variables which define the sanctions to be imposed for not following a rule.

Inspired by ADICO, we regard the specification of regulative norms as a process of identifying single regulative statements and their interrelations. As for single regulative statements, we specify a regulative norm as a combination of role and action that is obliged/forbidden before a deadline given a precondition, as formalized in Definition 5.3.

Definition 5.3 (Regulative norm) *A regulative norm n^r is defined as a tuple $(D, \rho, \delta, \sigma)$ where:*

- $D \in \{\mathrm{O}, \mathrm{F}\}$ *indicates the deontic type of the norm, i.e., Obliged, Forbidden,*
- $\rho \in Iv$, *describing a non-empty target to which the deontic modality is assigned,*

- $\delta \in L_\varepsilon$, *describing the deadline of the norm, and*
- $\sigma \in L_\varepsilon$, *describing the precondition of the norm.*

The target is an institutional event indicated by a role-action pair in which the role specifies to whom the norm applies and the action specifies what is directed and controlled by the norm. Both the precondition and the deadline are institutional event formulas (combination of role-action pairs). The precondition determines when the norm is activated, and the deadline determines when an obligation has to be ensured or when a prohibition ceases to be forbidden. Notice that if the obliged target is not ensured, at the moment that the deadline occurs a violation will be caused. As for a forbidden target, a violation is caused as soon as the target is brought about before the deadline occurs. If a norm does not specify a particular precondition, then the default interpretation is that it always holds. Without a particular deadline, the default value is the whole life cycle of an organization's regulation.

For example, we can model a regulation that "*If a student borrows a book from the library, the student should return the book within 1 month*" by defining a regulative norm n_1^r = (O, (*Student*, *return_book*), (*Timeline*, *p_1month*), (*Student*, *borrow_book*)).[1] In this regulative norm, we have defined two roles *Student* and *Timeline*, in which *Timeline* is a special role used to indicate the elapsing of time.

When a set of regulative norms are imposed to regulate agent behavior, they are usually interrelated in two ways: (1) they might pertain to the same role, constrain the same action, are activated under the same condition, are terminated by the same deadline, and (2) the compliance of a regulative norm might influence the compliance of other regulative norms and the compliance of each regulative norm contributes to the compliance of the organization as a whole.

The first type of relation is reflected by the fact that some elements are used in different regulative norms. For example, two regulative norms might refer to the same role, or the elements of the precondition of a regulative norm can be the deadline of another regulative norm. For the second type of relation, we consider three kinds of compliance relations between regulative norms. The first is that both regulative norms should be complied with and the violation of either regulative norm will result in a violation to the combination, represented as AND(n_1^r, n_2^r). The second indicates a choice between the two regulative norms and only when both regulative norms are violated the combination is considered as violated, represented as OR(n_1^r, n_2^r). The third indicates the two regulative norms are conditional and exclusive, i.e., (1) only when the first regulative norm is violated can the second regulative norm be activated, (2) the violation of the first regulative norm can be repaired by the fulfillment of the second regulative norm. This is represented as OE(n_1^r, n_2^r), and OE stands for Or Else.

To model a set of interrelated regulative norms, we introduce the concept of Regulative Norm Net, as shown in Definition 5.4.

[1] The numbering of the example norms is reserved for later references.

Definition 5.4 (Regulative norm net) *A regulative norm net RNN is defined by the following BNF:*

$$RNN{::} = n^r|\mathrm{AND}(RNN, RNN)|\mathrm{OR}(RNN, RNN)|\mathrm{OE}(RNN, RNN),$$

where n^r is a regulative norm; we use S_{RNN} to denote the set of all regulative norm nets contained in RNN and use Iv_{RNN} to denote the set of institutional events contained in RNN; when RNN is not a single regulative norm, we use LT_{RNN} to denote the left-hand side component of RNN, RT_{RNN} to denote the right-hand side component of RNN, CR_{RNN} to denote the compliance relation between the two components.

A regulative norm net RNN can be a single regulative norm n^r or a nested structure composed of regulative norms connected by AND, OR, and OE, which realizes a modular construction of interrelated regulative norms. For example, consider the following normative constraints "*students should submit their answer sheets within 5 minutes after the teacher announces the end of the exam, otherwise the teacher should revoke his or her qualification of the exam before the teacher marks the answer sheets.*" These normative constraints can be represented by a regulative norm net $RNN_1 = \mathrm{OE}(n_2^r, n_3^r)$ in which n_2^r = (O, (*Student*, *submit_answer*), (*Teacher*, *announce_end*)<(*Timeline*, *pass_5minute*), λ) and n_3^r = (O, (*Teacher*, *revoke_qualification*), (*Teacher*, *mark_answer*), λ).

5.1.3 Constitutive norms

Constitutive norms involve the creation of categories and the construction of typifications, by which concrete and subjectively unique experiences are subsumed under general orders of meaning that are both objectively and subjectively real [19]. In NNs, we use constitutive norms to interpret how events occurred in the physical world are perceived by organizations. Specifically, a constitutive norm indicates which institutional event is created when certain external events occur under specific conditions, as formalized in Definition 5.5.

Definition 5.5 (Constitutive norm) *Let a constitutive norm n^c be a mapping function $n^c : (\beta, \kappa) \to \nu$, representing a counts-as relation, where*

- *$\beta \in L_e$ is an external event formula such that $\forall$ (ag_i, α_i), $(ag_j, \alpha_j) \in E_\beta : ag_i = ag_j$,*
- *$\kappa \in L_e$ is an external event formula, indicating the condition where β occurs, and*
- *$\nu \in Iv$ is an institutional event, indicating the institutional occurrence perceived by the organization.*

By constitutive norms, we are able to derive institutional consequences from the occurrence of specific external events in the physical world. That is, a combination of agent actions together *counts as* an action of a role in the organization. Moreover, if β contains more than one event, all the events should be related to the same agent so that the responsibility can be identified. An example of constitutive norm could be $n_1^c = ((Alice,\ input\_barcode){<}(Alice,\ swipe\_card),\ (Alice,\ enter\_buildingL)){\to}(Student,\ borrow\_book)$, indicating the occurrence "*after Alice enters the building L, Alice first inputs a barcode and then swipes her card*," counts as "*a student borrows a book*."

Up to now, we have introduced the formalisms to capture both the regulative and constitutive aspects of compliance rules, as well as the compliance relations between them. In the next subsection, we combine all these formalisms and introduce the construction of a Normative Structure.

5.1.4 Norm Nets

A normative structure is intended to model a set of compliance rules, considering both the regulative and constitutive aspects of the rules as well as the compliance relations between the rules. To this end, we combine the formalisms of regulative norm net and constitutive norm presented earlier, and introduce the concept of *Norm Net* in Definition 5.6.

Definition 5.6 (Norm net) *A norm net NN is defined as a tuple (RNN, CNN) where*

- *RNN is a regulative norm net built over a set of regulative norms,*
- *CNN, named constitutive norm net, is a set of constitutive norms.*

We use Ev_{NN} to denote the set of external events defined in NN and Iv_{NN} to denote the set of institutional events defined in NN.

Each norm net defines a set of regulative norms and their compliance relations so that the behavior of participating agents can be collectively regulated. The set of constitutive norms presents an interpretation of the possible occurrences in the physical world. In this way, Norm Net provides a formalism for the construction of the normative structure in OperA+. Note that we use the terms Normative Structure NS and Norm Net NN interchangeably in this dissertation.

Now that we have given the definitions of a social structure and a normative structure, we can see that the link between the social structure and normative structure of an organizational model is the set of roles specified in the social structure and the set of norms defined over the actions of the roles.

Up to this point, we have presented the formalism for the construction of the normative structure of organizational models. In the next section, we investigate the dynamic aspects of the normative structure, i.e., how norms evolve subjected to the actions of the agents being regulated.

5.1.5 State Transition of Norms

Before discussing the states of norms, we differentiate between two concepts: norm specifications and norm instances. *Norm specifications* can be seen as templates for creating norm instances and these templates do not change along with the occurrences of agent actions. The definitions presented in the previous sections, are the syntax for creating norm specifications. *Norm instances* represent the dynamic aspect of norms, resulting from the consequence of agent actions. These actions can lead to the activation of some norm instances and also to the compliance or violation of the norm instances. To indicate such changes, we define *normative states* to describe the effects of agent actions on norm instances. In this sense, an instance of a norm specification also incorporates a normative state, as captured in Definition 5.7.

Definition 5.7 (Norm instance) *An instance of a norm net* NN *is defined as* $NN^i = (NN,\ s)$ *where*

- NN *is a norm net specification as described in Definition 5.6, and*
- s *indicates the normative state of* NN^i*.*

An instance of a norm net consists of the specification of the norm net and the normative state of the norm net instance. Similarly, we can have an instance of a regulative norm, a constitutive or a regulative norm net. A norm specification can have multiple instances and each of the instances have its own normative state. It has to be noticed that, when talking about states, we always refer to norm instances but may omit the word "instance" to simplify the description.

In our setting, constitutive norms determine the creation of certain institutional events in an organization given the occurrence of some external events, and these institutional events further lead to the changes of the normative state of the regulative norms specified by the organization. When an instance of a regulative norm is created, the normative state of the instance is set to be *instantiated*, indicating its participation in the control of agent behavior. Given an instantiated regulative norm, as soon as the precondition in the norm is fulfilled, the state of the norm transits from *instantiated* to *active*. Being active means that the norm begins to take effect in terms of its deontic type. From the active state, a regulative norm can move into one of the two states: *satisfied* and *violated*, and the change of the normative state is differentiated between obligations and prohibitions, illustrated respectively in Figure 5.1 (a) and (b). For an obligation, when the obliged action is not performed by the agents enacting the specified role before the deadline of the norm occurs, the norm transits to a *violated* state, otherwise to a *satisfied* state when the action is performed. On the other hand, a prohibition will transit to a *violated* state if the forbidden action is performed before the deadline occurs. It moves to a *satisfied* state if the deadline occurs and the forbidden action has not been performed before. For example, when creating an instance from the obligation n_1^r = (O, (*Student*, *return_book*), (*Timeline*, *p_1month*), (*Student*, *borrow_book*)), the instance is instantiated. If the precondition (*Student*, *borrow_book*) occurs, the norm instance moves into the active state with the target (*Student*, *return_book*) being obliged. At this moment, when the target is brought about before the deadline (*Timeline*, *pass_1month*) is fulfilled, the norm instance transits into the satisfied state, otherwise it transits into the violated state when the deadline occurs.

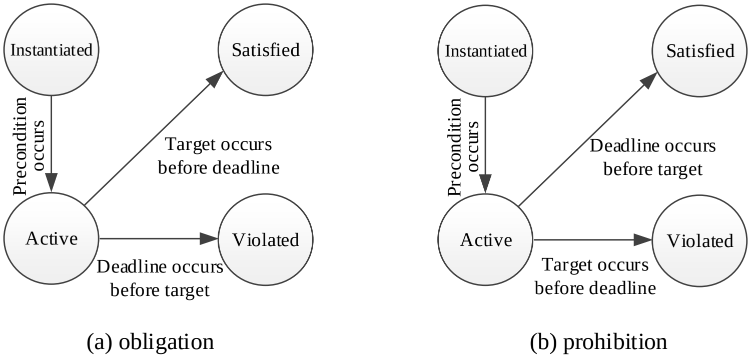


Figure 5.1: *Normative state transitions of a regulative norm instance.*

Each instance of a regulative norm can be in one of these four exclusive normative states, i.e., instantiated, active, satisfied, and violated. To indicate the compliance status of a regulative norm instance, we introduce Compliance Evaluation State (*CES*) as a projection of the normative states of the regulative norm. The *CES* of a regulative norm can have two values: compliant and violated. The *compliant* state indicates that the regulative norm is not in a violated state, which is weaker than a satisfied state as it also includes the situation where the norm instance is in an instantiated or active state and has not (yet) transited into a satisfied or violated state. That is, the compliant state is defined as a union of the instantiated, active and satisfied states.

The impact of constitutive norms on the normative states is reflected in their ability of generating institutional events that constitute the targets, preconditions and deadlines in regulative norms. When a constitutive norm instance is created, it will stay active and produce institutional events as soon as the corresponding external events occur. For example, with an stance creating from the constitutive norm n_1^c = ((*Alice*, *input_barcode*)<(*Alice*, *swipe_card*), (*Alice*, *enter_buildingL*))→(*Student*, *borrow_book*), when there is an occurrence sequence (*Alice*, *enter_buildingL*)<(*Alice*, *input_barcode*)<(*Alice*, *swipe_card*) in the physical world, an institutional event (*Student*, *borrow_book*) will be generated in the organization. Moreover, the generation of the institutional event (*Student*, *borrow_book*) indicates that the precondition of n_1^r is fulfilled, which will activate the instance of n_1^r accordingly.

The normative state of a regulative norm net is obtained by a collective evaluation of its consisting regulative norms with respect to their compliance relations. That is, to determine the normative state of a regulative norm net, the normative state of all the consisting regulative norms in the regulative norm net have to be evaluated and combined based on their compliance relations. The combined evaluation is defined in Table 5.1.

compliance relation	AND				OR				OE			
left \ right	I	A	S	V	I	A	S	V	I	A	S	V
Instantiated(I)	I	A	A	V	I	A	S	A	I	/	/	/
active (A)	A	A	A	V	A	A	S	A	A	/	/	/
satisfied (S)	A	A	S	V	S	S	S	S	S	/	/	/
violated (V)	V	V	V	V	A	A	S	V	A	A	S*	V

Table 5.1: *Resulting normative states of a regulative norm net consisting of two sub-regulative norm nets with different compliance relations*

With an AND relation, as long as one of the component regulative norm nets is in a violated state the normative state of the combined regulative norm net is considered to be violated, while only when both component regulative norm nets are in a satisfied state the normative state of the combined regulative norm net is considered to be satisfied. When both component regulative norm nets are in an instantiated state, the combined regulative norm net is considered to be in an instantiated state. The rest of the combinations lead to the combined regulative norm net being active.

With an OR relation, as long as one of the component regulative norm nets is in a satisfied state the normative state of the combined regulative norm net is considered to be satisfied, while only when both component regulative norm nets are in a violated state the normative state of the combined regulative norm net is considered to be violated. When both component regulative norm nets are in a instantiated state, the combined regulative norm net is considered to be in an instantiated state. The rest of the combinations lead to the combined regulative norm net being active.

With an OE relation, we distinguish between the two component regulative norm nets by considering the left-hand regulative norm net as the *origin* and the right-hand regulative norm net as the *reparation* to the violation of the origin. When the origin is in an instantiated, active or satisfied state, the reparation can only be in an instantiated state since only when the origin is violated can the reparation be activated. In these cases, the normative state of the combined regulative norm net is considered to be the same as that of the origin. When the origin is in a violated state and the reparation is in an instantiated state, the normative state of the combined regulative norm net is considered to be in an active state. For the rest of the combinations in which the origin is in a violated state, the normative state of the combined regulative norm net stays the same as that of the reparation. However, in case that the satisfied state of the combined regulative norm net is derived from the combination of the origin being violated and the reparation being satisfied, this satisfied state is distinguished since there has been a repaired violation. To this end, we use the symbol S^* to represent such a state and we call it a *sub-ideal satisfied state*.

Each instance of a regulative norm net can be in one of the five normative states, i.e., instantiated (I), active (A), satisfied (S), violated (V) and sub-ideal satisfied (S^*). Similarly to that of a single regulative norm instance, we use *CES* to indicate the compliance status of a regulative norm net. The difference is that when the *CES* of a regulative norm net is compliant, the normative state of the regulative norm net may be S^* besides I, A and S. Based on the changes of normative states shown in Table 5.1, we derive the changes of *CES* accordingly in Table 5.2.

compliance relation	AND		OR		OE	
right / left	C	V	C	V	C	V
compliant (C)	C	V	C	C	C	C
violated (V)	V	V	C	V	C^*	V

Table 5.2: *Resulting compliance evaluation state of a regulative norm net consisting of two sub regulative norm nets with different compliance relations*

It can be seen that, with an AND relation, only when the *CES* of both component regulative norm nets are compliant the *CES* of the combined regulative norm net is compliant, while as long as the *CES* of one of the component regulative norm nets is violated the *CES* of the combined regulative norm net is violated. The resulting *CES* of the combined regulative norm net with an OR relation is a reverse of that of the combined regulative norm net with an AND relation. The resulting *CES* of the combined regulative norm net with an OE relation is similar to that of the combined regulative norm net with an OR relation, except that when the *CES* of the origin is violated and the *CES* of the reparation is compliant the *CES* of the combined regulative norm net is considered to be a *sub-ideal compliant* denoted C^* (corresponding to the sub-ideal satisfied state S^* in Table 5.1).

A norm net is composed of a regulative norm net and a constitutive norm net. Above, we have explained the changes of the *CES* of a regulative norm net. As for a constitutive norm net, since the functionality of constitutive norms is generating institutional events according to the occurrence of external events, the constitutive norm net mainly serves as an interpreter to the changes of the *CES* of regulative norm nets. In this sense, the *CES* of a norm net can be seen as equivalent to the *CES* of its consisting regulative norm net, i.e., the *CES* of a norm is either compliant or violated according to the *CES* of its consisting regulative norm net.

5.2 Operational Semantics

To give operational semantics to NNs and facilitate compliance checking of agent actions, we propose a mapping from NNs to Colored Petri Nets (CPNs) [112]. There are mainly two reasons of choosing CPNs for operational semantics. The first reason is its ability of describing and analyzing concurrent processes. The concurrent aspect of NNs lies in the normative system that NNs is modeling, where agents take actions concurrently and their behavior has to be evaluated concurrently. The second reason is the availability of well-founded techniques and tool support of CPNs for simulation and verification. This is important for our purpose of implementing compliance evaluation, which requires both a simulation and verification means. In the remainder of this section, we illustrate how the components of NNs are mapped to CPNs. For an overview of CPNs, we refer the reader to the introduction given in Chapter 2.

5.2.1 Mapping of Single Regulative Norms

The basic elements of a regulative norm are institutional events indicated by role-action pairs. For each institutional event, the *role* maps to a *color set* and the *action* maps to a *transition* in CPNs. Each place in a CPN is assigned a specific color set, and tokens situated in the place represent agents who play corresponding roles. That is, a role is represented by a data type while an agent enacting that role is of the corresponding data type. Each transition in a CPN represents an action that role enacting agents may perform. The connections between places and transitions indicate which actions are relevant to which roles and whether the actions are performed by the agents enacting the roles.

Given a regulative norm $n^r = (D, \rho, \delta, \sigma)$, the construction of its CPN model follows three steps:

1. constructing CPN snippets for each institutional event in the construction of the target ρ, the deadline δ and the precondition σ (see Figure 5.2),

2. combining the CPN snippets obtained from the first step according to the relations ($\wedge$, $\vee$, $<$) between the corresponding institutional events in the precondition σ and the deadline δ (see Figure 5.3, 5.4 and 5.5), and

3. connecting the combined CPN snippets obtained from the second step to obtain the CPN model of a regulative norm according to the deontic type of the norm (see Figure 5.6 and 5.7).

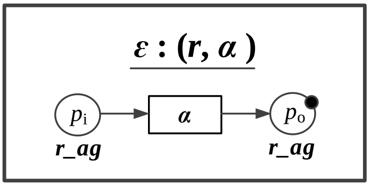


Figure 5.2: *CPN pattern for an institutional event.*

An institutional event (r, α) is represented by a place connecting to a transition which connects to another place, as shown in Figure 5.2. Both places are associated with a color set r_ag indicating that agents represented by the residing tokens are enacting the role r. The transition refers to the action α (indicated by the name of the action). The input place p_i of α represents the state when the event (r, α) does not occur while the output place p_o represents the state when the event (r, α) occurs. Whether or not there is an occurrence of the event (r, α) is indicated by the existence of tokens in the place p_o. It has to be noticed that an event can occur multiple times. Therefore, the number of tokens in p_o also indicate the number of occurrences of that event. For example, a token, denoted as a black dot, is attached to the output place p_o, indicating that there has been an occurrence of the event (r, α). Notice that the "color" of a token is only a data-type/label, not per se an actual color (black, green, red, etc.).

Institutional event components	**CPN components**	**Example**
role r	color set $r = with\ r_{name}$	*student*
agents Ag	color set $Agent = string$	*"Bob"*
agents enacting a role r_ag	color set $r\_ag = product\ r * Agent$	(*student*, *"Bob"*)
action α	transition α	$\alpha = borrow\_book$
state of occurrence (r, α)	places p_i, p_o	

Table 5.3: *Correspondence between the components in an institutional event and the components in a CPN model*

Table 5.3 summarizes the mapping between the components in an institutional event and the components in a CPN model. Each role is defined by an enumeration color set with a single value, i.e., the name of the role. The color set Agent is defined by the simple type *string*, whose values are agent names. To indicate the agents enacting a specific role, the color set r_ag is defined as a combination of the role and agents through the type *product*. The reason of using a constant to represent a role is to make sure that only tokens representing the right agents can enter a place, i.e. the agents who enact the role assigned to the place can get access to the place. Such assignments of roles to places are enabled by labeling the places with the corresponding color set *r_ag*. Algorithm 1 shows the procedure of generating the CPN model given an institutional event.

Algorithm 1 CPN mapping of an institutional event: InsEvent_CPN

Require: ε ▷ An institutional event
Ensure: $\mathcal{N}$ ▷ A CPN model
$(r, \alpha) \leftarrow \varepsilon$
 ▷ (1) Create a transition representing the action α in the event
$T_{\mathcal{N}} \leftarrow \{\alpha\}$

 ▷ (2) Create two places representing the state whether the event occurs
$P_{\mathcal{N}} \leftarrow \{p_i, p_o\}$
$A_{\mathcal{N}} \leftarrow \{(p_i, \alpha), (\alpha, p_o)\}$

 ▷ (3) Create color sets and assign them to the places
colset Agent = string ▷ create a color set representing agents
colset r = with name$_r$ ▷ create a color set representing a role
colset r_ag = product r ∗ Agent ▷ create a color set representing agents enacting a specific role
$\Sigma_{\mathcal{N}} \leftarrow \{Agent, r, r\_ag\}$
$C_{\mathcal{N}} \leftarrow \{(p_i, r\_ag), (p_o, r\_ag)\}$

For example, given the target (*Student*, *return_book*) in n_1^r, we can create a CPN model following Algorithm 1. The color of the places is defined by product of the role *Student* and *Agents*, and the transition is defined by the action *return_book*. The place p_i indicates the state when the event (*Student*, *return_book*) does not occur while the place p_o indicates the state when the event (*Student*, *return_book*) occurs.

The combination of institutional events is realized through the three operators in L_E: $\wedge$ (and), $\vee$ (or), $<$ (before). Based on the workflow patterns presented in [181], the CPN patterns for the three types of combination are described as follows.

(1) $\wedge$ **relation:** Figure 5.3 shows the CPN pattern for two institutional event formulas ψ_1 and ψ_2 combined with a $\wedge$ relation which indicates that only when both ψ_1 and ψ_2 occur the combination of these two is seen as occurred. Accordingly, the sink places of both branches are connected to the same place through the same transition such that the thread of control is converged only when both input branches have a token in their sink places.

(2) $\vee$ **relation:** Figure 5.4 shows the CPN pattern for two institutional event formulas ψ_1 and ψ_2 combined with a $\vee$ relation which indicates that as long

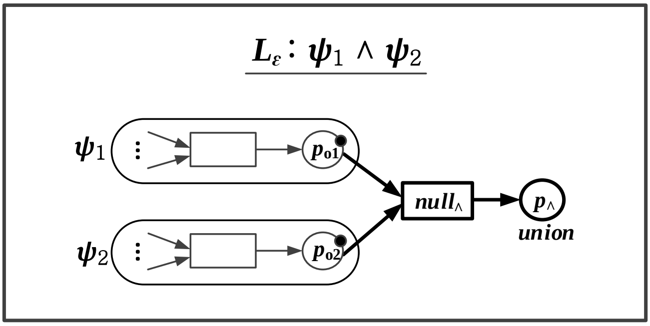


Figure 5.3: *CPN pattern for the combination of institutional events with an ∧ relation.*

as one of ψ_1 and ψ_2 occur the combination of these two is seen as occurred. Accordingly, the sink places of both branches are connected to the same place through two different transitions such that the thread of control is converged when either input branch has a token in its sink place.

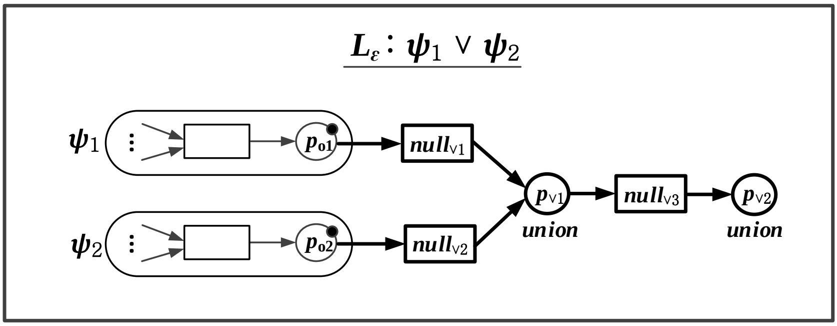


Figure 5.4: *CPN pattern for the combination of institutional events with an ∨ relation.*

For both $\wedge$-relation and $\vee$-relation, all the places in the succeeding branch are assigned a new color set, being the union of the color set of the sink places in the two preceding branches. This is to enable that agents playing any of the roles in the union can move to those places.

(3) $<$ **relation:** Figure 5.5 shows the CPN pattern for two institutional event formulas ψ_1 and ψ_2 combined with a $<$ relation which indicates that only when ψ_1 occurs first and then ψ_2 occurs the combination of these two is seen as occurred. To achieve this, the sink place of the first branch is connected to the output transitions of all the source places in the second branch such that the thread of control is passed to the second branch only when the first branch has a token in its sink place. Since there might be multiple source places in the CPN model of ψ_2, the token in the sink place of ψ_1 is processed into multiple copies such that the output transition of each source place in ψ_2 can be enabled when ψ_1 occurs.

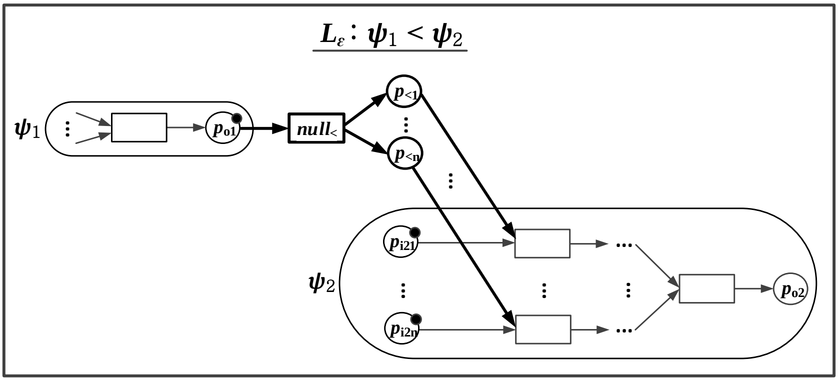


Figure 5.5: *CPN pattern for the combination of institutional events with an $<$ relation.*

Notice that there is a set of transitions labeled with *null* in Figures 5.3 and 5.4. These transitions are necessary in order to apply the synchronization mechanism of CPNs, and they will fire as soon as their input places have sufficient tokens. A formal procedure of generating the CPN model of a given institutional event formula is presented in Appendix A.3.

Suppose that we have two institutional events $\psi_1 = (Student, borrow\_book)$ and $\psi_2 = (Librarian, register\_book)$. For each of these two events, we can build a CPN model following the mapping rule of a single institutional event shown in Figure 5.2. If the two events are combined with an $\wedge$ relation, we can connect their CPN models accordingly to express that only when both events occur (producing a token in both $p_o^{\psi_1}$ and $p_o^{\psi_2}$) the compound event $\psi_1 \wedge \psi_2$ is seen as occurred (automatically producing a token in the sink place $p_\wedge$). If an $\vee$ relation is assigned to them, the occurrence of either of the two events (a token is produced in either $p_o^{\psi_1}$ or $p_o^{\psi_2}$) indicates the occurrence of the compound event $\psi_1 \vee \psi_2$ (automatically producing a token in the sink place $p_{\vee 2}$). If the two events are with an $<$ relation, the occurrence of ψ_1 (a token is produced in $p_o^{\psi_1}$) will enable ψ_2 to happen, and only when ψ_2 occurs the compound event $\psi_1 < \psi_2$ is seen as occurred (producing a token in the sink place $p_o^{\psi_2}$).

The combination of the target, precondition and deadline of a regulative norm should follow the logic that after the occurrence of the precondition the target should (not) occur before the occurrence of the deadline otherwise a violation is generated. The precondition determines when the regulative norm is activated while the deadline determines when the regulative norm can be evaluated to be satisfied or violated (depending on the deontic type of the norm). To represent such correlations in a regulative norm, the construction of its CPN model follows two steps (see Figure 5.6 and 5.7):

(1) activation: connecting the sink place of the precondition to the transition of the target and to the input transition of the sink place of the deadline, such that only when the precondition is fulfilled the target and the deadline are enabled to be evaluated.

(2) evaluation: connecting the source place of the target to the input transition of the sink place of the deadline, such that when the deadline occurs the responsible token representing an agent can be transferred to the satisfied or violated place according to the deontic type of the regulative norm.

Figure 5.6 and 5.7 respectively shows the CPN pattern of an obligation and a prohibition. The active, satisfied and violated states of a regulative norm are indicated by the three places labeled A, S and V in the CPN model of the regulative norm. These three places are respectively the sink place in the CPN patterns of the precondition, target and deadline of the norm, indicating the state when the corresponding events occur. The instantiated state (I) is indicated by the combination of all the rest of the places in the CPN pattern. Moreover, both the S and V places are assigned a color set REA defined as a list of the union of the color sets representing all the role-enacting agents regulated by the norm specification. In the case of a single obligation or prohibition, REA is a list of the union of the color set r_ag derived from the target.

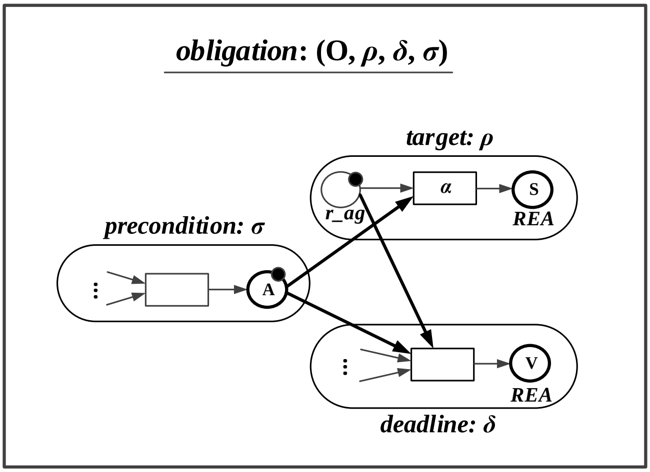


Figure 5.6: *CPN pattern for an obligation.*

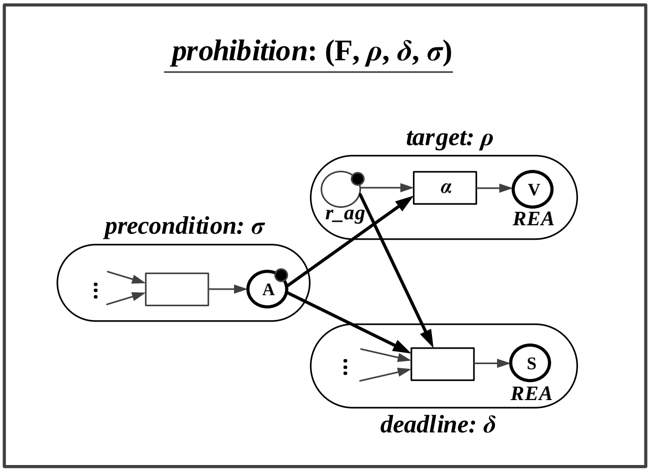


Figure 5.7: *CPN pattern for a prohibition.*

The differences between an obligation and a prohibition are that

- the firing of the obliged transition (target) will lead to the obligation being satisfied (S) while the firing of the forbidden transition (target) will lead to the prohibition being violated (V), and
- the firing of the transitions corresponding to the deadline will lead to the obligation being violated (V) while lead to the prohibition being satisfied (S).

As such, when a token appears in those featured places (labeled A, S, V), we know in which normative state an instance of a regulative norm is situated. Specifically, when there is no token in the violated place, the norm instance is considered to be in a compliant state (either in an instantiated, active or satisfied state). A formal procedure of generating the CPN model of a given regulative norm (obligation or prohibition) is presented in Appendix A.4.

For example, to obtain the CPN model of n_1^r, we first build the CPN model for the target (*Student*, *return_book*), the precondition (*Student*, *borrow_book*), and the deadline (*Timeline*, *pass_1month*) following the mapping rules in Figure 5.2 and 5.5, and then connect them according to the mapping rule in Figure 5.6. When the event (*Student*, *borrow_book*) occurs, a token is produced in the sink place of the CPN model of the precondition, with n_1^r being activated and the event (*Student*, *return_book*) being obliged. However, without the occurrence of (*Student*, *borrow_book*), even if (*Student*, *return_book*) occurs or (*Timeline*, *pass_1month*) occurs, we cannot determine whether n_1^r is satisfied or violated.

5.2.2 Mapping of Regulative Norm Nets

Given a regulative norm net, a CPN model is built for each regulative norm in the regulative norm net. If two regulative norms have shared components, they will have some shared constructs in their CPN models. Moreover, we need to represent the compliance relations between the regulative norms in the regulative norm net. In Section 5.1.5, we have shown the compliance states of regulative norm nets with different compliance relations, summarized in Table 5.1. On that basis, we present how the CPN models of regulative norms are connected with respect to the compliance relations they have.

(1) AND(RNN_1, RNN_2)**:** There are two aspects to be represented in the combination with an AND relation. Firstly, the violation of either of the two component regulative norm nets can lead to the combination being violated. Accordingly, we connect the violated place of each component regulative norm net (labeled V) to an overall violated place through a different transition such that as soon as there is a token in one of the violated place of the component regulative norm nets, a token will be produced in the overall violated place, as shown in Figure 5.8. Secondly, only when both component regulative norm nets are satisfied, the combination can be seen as satisfied. Accordingly, we connect the satisfied places of both component regulative norm nets (labeled S) to an overall satisfied place through a single transition such that only when there are tokens in the satisfied place of both component regulative norm nets, a token will be produced in the overall satisfied place.

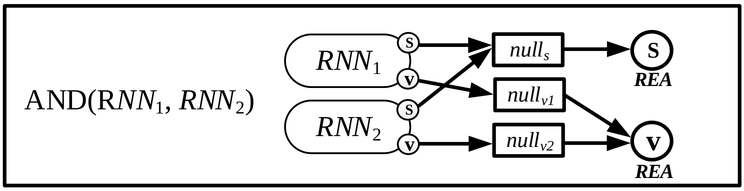


Figure 5.8: *CPN pattern of two regulative norms nets combined with an AND relation.*

(2) OR(RNN_1, RNN_2)**:** Similarly, there are two aspects to be represented in the combination with an OR relation. Firstly, only when both component regulative norm nets are violated, the combination can be seen as violated. Accordingly, we connect the violated places of both component regulative norm nets (labeled V) to an overall violated place through a single transition such that only when there are tokens in both the violated places of the component regulative norm nets, a token will be produced in the overall violated place, as shown in Figure 5.9. Secondly, the satisfaction of either of the two component regulative norm nets can lead to the combination being satisfied. Accordingly, we connect the satisfied place of each component regulative norm net (labeled S) to an overall satisfied place through a different transition such that as soon as there is a token in either satisfied place of the component regulative norm nets, a token will be produced in the overall satisfied place.

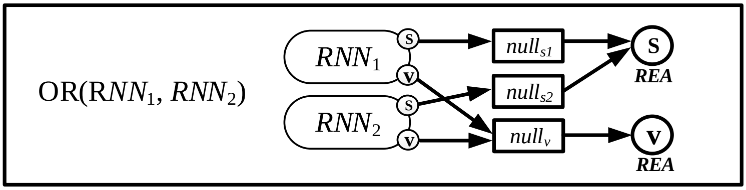


Figure 5.9: *CPN pattern of two regulative norms nets combined with an OR relation.*

(3) OE(RNN_1, RNN_2)**:** There are four aspects to be represented in the combination with an OE relation. Firstly, the satisfaction of either of the two component regulative norm nets can lead to the combination being satisfied. Accordingly, the satisfied place of each component regulative norm net is connected to the overall satisfied place through two different transitions, as shown in Figure 5.10. Secondly, only when the reparation is violated the combination can be seen as violated. Accordingly, only the violated place of RNN_2 is connected to the overall violated place. Thirdly, the violation of RNN_1 should be distinguished since this violation may be repaired. Accordingly, we make a record of the violation of RNN_1 by a place labeled *vr*. Fourthly, only when RNN_1 is violated can RNN_2 be activated, indicating a conditional relation between the two. Accordingly, the place *vr* in RNN_1 is bidirectionally connected to the output transitions of all the source places in RNN_2. As a result, only with a token in the violation recording place *vr* of RNN_1 can RNN_2 be activated. It should be noted that we use bidirectional arcs for the connections of the place *vr* because we want to keep the token in *vr* as a record of the violation of RNN_1.

Similarly, there are two sink places labeled as S and V in the CPN construction of a regulative norm net, respectively representing the satisfied and violated state of the regulative norm net. These two sink places S and V are also assigned a color set REA, representing all the role-enacting agents regulated by the entire norm specification. When there is a token in the place S (or V), the instance of the regulative norm net is considered to be in a satisfied (or violated) state. On the other hand, when neither of these two places has a token, the instance of the

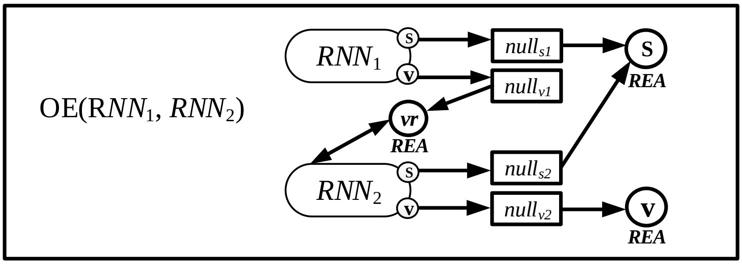


Figure 5.10: *CPN pattern of two regulative norms nets combined with an OE relation.*

regulative norm net is considered to be in a compliant state. In this way, we can combine a set of regulative norm nets into a larger regulative norm net according to their compliance relations, while preserving the interface of one satisfied place and one violated place in the resulting CPN model. Moreover, there is a set of places labeled vr which are used to indicate the violations that could be repaired. Based on the description above, a formal procedure of generating the CPN model of a given regulative norm net is presented in Appendix A.5.

5.2.3 Mapping of Constitutive Norms

Constitutive norms, represented as counts-as rules, create a link between agents' actions in the physical world (represented by external event formulas) and the effects perceived by the organizations (represented by institutional events). For each constitutive norm, a combination of agent actions is mapped to an action of a role. Therefore, to obtain the operational semantics of a constitutive norm, we have to capture the relation between the actions of the agents and the action of the role, i.e., how the occurrence of some external events are interpreted as the occurrence of an institutional event. For this purpose, we adopt the model of hierarchical CPNs introduced in Chapter 2. The idea is that the institutional occurrence (the right-hand side component of a counts-as rule) can be seen as a prime module which will be triggered by a sub module that represents the conditional occurrence in the physical world (the left-hand side component of a counts-as rule).

Given a constitutive norm $(\beta, \kappa) \to \nu$, the construction of its CPN model has the following steps.

1. build a CPN model for ν using the mapping rule shown in Figure 5.2, reserving for a prime module of a hierarchical CPN model (as defined in Definition 2.4 in Chapter 2),

2. respectively build a CPN model for β and κ by adapting the mapping rule shown in Figure 5.2 to 5.5: all the places in the CPN model are assigned

the color set *Agent* and the input place is assigned a token with the value of the agent's name,

3. copy both the input and output place of the CPN model of ν and connect them by a *null* transition (without an corresponding action and will fire immediately when enabled), and connect the sink place of the CPN model of β to that transition.

4. integrate the CPN model of β and κ by $<$-relation to capture the conditional relation between the two external event formulas, serving as a submodule of the hierarchical CPN model,

5. build a module hierarchy: (1) replacing the transition of the prime module obtained in the first step with a substitution transition, (2) assigning the submodule obtained from the fourth step to the substitution transition by respectively relating the input and output places of the substitution transition with their copies made in the third step via port-socket relations.

A graphical illustration of the CPN mapping of a constitutive norm is shown in Figure 5.11 and a formal procedure of generating a hierarchical CPN model given a constitutive norm is presented in Appendix A.6.

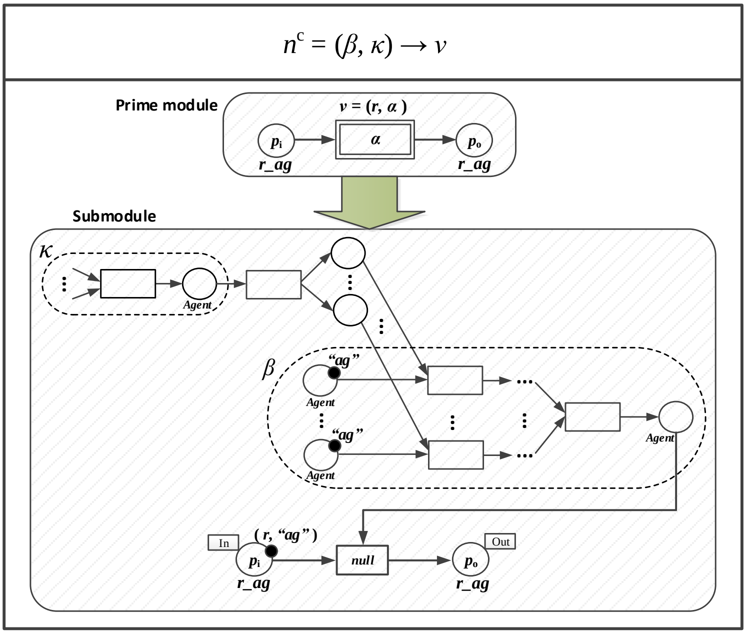


Figure 5.11: *CPN pattern of a constitutive norm*

As described in Definition 5.6, a norm net contains a set of constitutive norms. It is possible that some of the constitutive norms might have the same institutional event as their right-hand side component ν. If this is the case, we combine all the left-hand side components of the constitutive norms by an $\vee$-relation to indicate that the physical occurrence (β, κ) of each of the constitutive norms can generate an institutional event ν.

As an example, we show the CPN model of the constitutive norm n_1^c in Figure 5.12. The institutional event (*Student*, *borrow_book*) is indicated by a

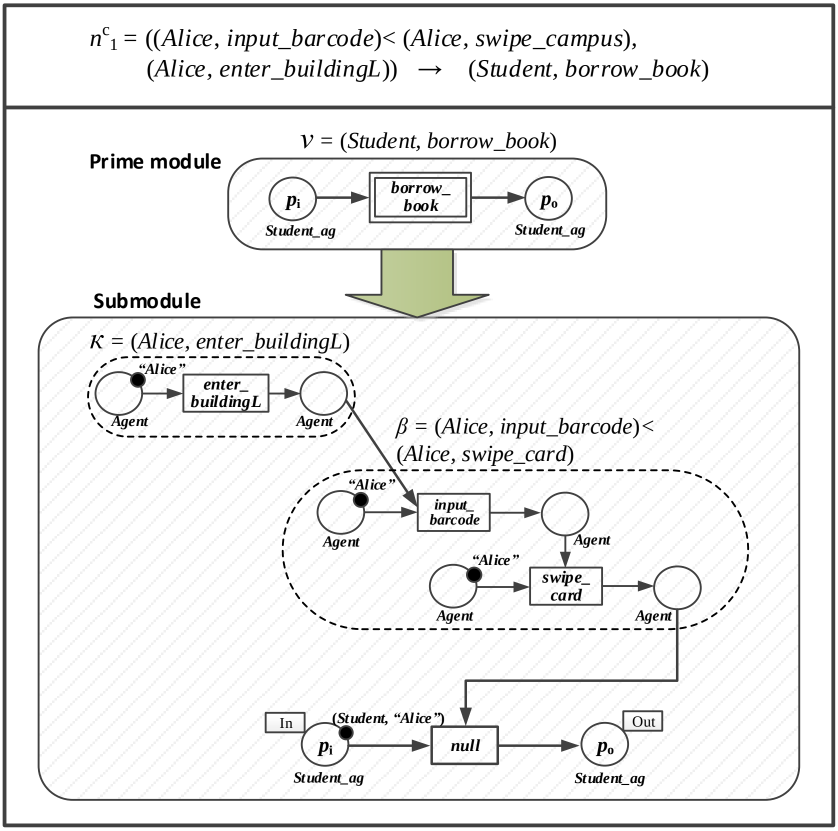


Figure 5.12: *CPN model of an example constitutive norm*

prime module containing a substitution transition *borrow_book* and two places colored *Student*. The substitution transition refers to a submodule representing the conditional occurrence of an external event in the physical world, i.e., ((*Alice*, *input_barcode*)<(*Alice*, *swipe_card*), (*Alice*, *enter_buildingL*)). The connection between the prime module and the submodule characterizes the "counts as" relation of the constitutive norm n_1^c. In the submodule, there are two port places which are labeled p_i and p_o. These two places are respectively related with the place having the same label in the prime module via port-socket relations. The institutional event (*Student*, *borrow_book*) will occur in the organization when the agent *Alice* first performs the action *enter_barcode* and then the action *wipe_card* given that the agent *Alice* has already taken the action *enter_buildingL* in the physical world. Accordingly, in the CPN model, if there is a firing sequence ⟨ (*Alice*, *enter_buildingL*), (*Alice*, *input_barcode*), (*Alice*, *swipe_card*) ⟩ in the submodule, the substitution transition *borrow_book* in the prime module is fired.

The link between the actions in the physical world and the actions perceived by the organization is indicated by the substitution transition in the prime module and all the transitions in the submodule. For the link between the roles specified in the organization and the agents governed by the organization, we assume that we can always determine the role assignment based on an agent's actions, e.g., killer (based on the occurrence of the action of killing), or manager (based on a predefined social structure of an organization).

5.2.4 Mapping of Norm Nets

To obtain the CPN model of a norm net $NN = (RNN, CNN)$, the following steps are needed:

1. build a CPN model for RNN based on the mapping rules described in Section 5.2.1 and 5.2.2.

2. build a hierarchical CPN model for each constitutive norm in CNN based on the mapping rules described in Section 5.2.3.

3. group all the hierarchical CPN models obtained from the second step into different module hierarchies according to their prime module.

4. for each module hierarchy obtained from the third step, search in the CPN model of RNN for all the CPN snippets that match the prime module in the module hierarchy.

5. for each of the matched snippets obtained from the fourth step, replace the transition in the snippets with a substitution transition and assign to it the submodules of the corresponding prime module.

In Figure 5.13, we show the CPN model of a norm net consisting of the regulative norm n_1^r and the constitutive norm n_1^c presented earlier. The prime module is obtained from the CPN model of n_1^r while the submodule is obtained from the CPN model of n_1^c. The submodule is assigned to the substitution transition labeled *return_book* in the prime module according to the counts-as relation specified in n_1^c.

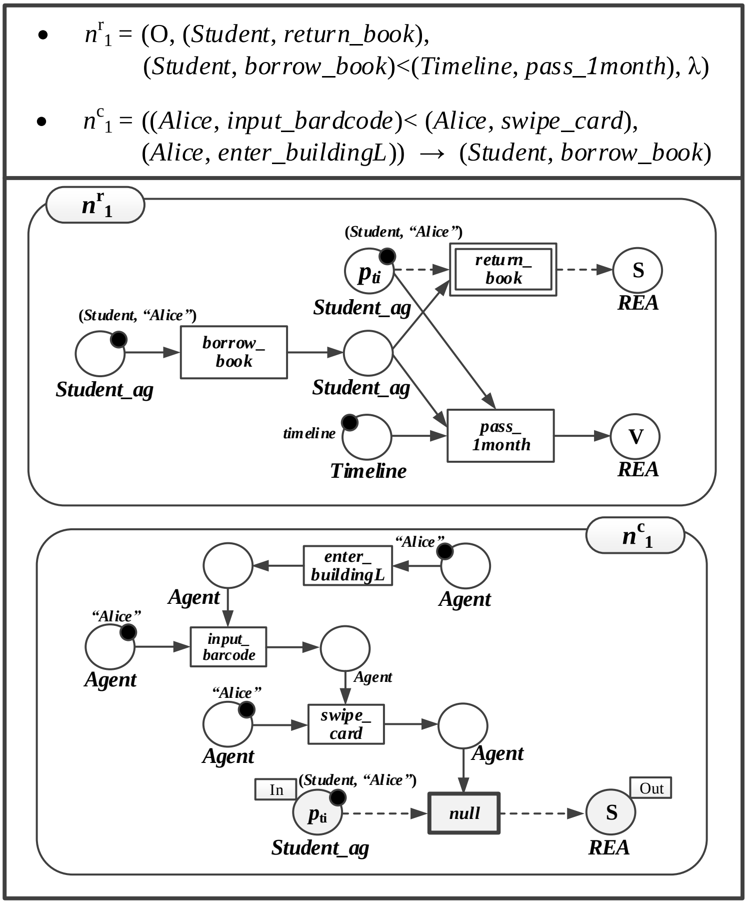


Figure 5.13: *CPN model of an example norm net*

In the mapping rules of both regulative and constitutive norms (see from Figure 5.3 to 5.11), we have used a group of transitions (indicated by the name *null*) to represent the structural relations in NNs. These transitions are those without corresponding actions and will fire immediately when they are enabled. In order to show which transitions refer to the same action and will fire simultaneously, we use a naming function $f_{\text{ACTION}}(t) = \alpha$ to associate a transition t with the action α. That is, given a transition by its name or label, this naming function will return the action that the transition represents. As a convention, we use the same prefix in the labels/names of the transitions that represent the same action.

5.2.5 Mapping of Normative States

In CPNs, markings (M) are used to represent the state of the system being modeled. Therefore, normative states of norm instances can be captured by the markings of the corresponding CPN models. Now that we have both the mapping relation between norm specifications and CPN constructions, and the mapping relation between normative states and CPN markings, instances of norm nets can be fully captured by CPN models. To indicate the resulting CPN model of a norm net instance, we give the definition of a CPN mapping as follows.

Definition 5.8 (CPN mapping of NNs) *The CPN mapping of a norm net instance NN^i is denoted as $\Theta(NN^i) = \langle(\mathcal{N}_H, P_s, P_v, P_{vr}, p_{so}, p_{vo}), M\rangle$ where,*

- *$\mathcal{N}_H$ is a hierarchical CPN model, according to Definition 2.5,*
- *$P_s \in P_{\mathcal{N}_H}$ is the set of places whose labels start with s from all the places in $\mathcal{N}_H$, indicating all the satisfied states of NN^i,*
- *$P_v \in P_{\mathcal{N}_H}$ is the set of places whose labels start with v from all the places in $\mathcal{N}_H$, indicating all the violated states of NN^i,*
- *$P_{vr} \subseteq P_{\mathcal{N}_H}$ is the set of places whose labels start with vr from all the places in $\mathcal{N}_H$, indicating all the repairable violation states of NN^i,*
- *$p_{so} \in P_{\mathcal{N}_H}$ is a sink place labeled by s, indicating the overall satisfied state of NN^i,*
- *$p_{vo} \in P_{\mathcal{N}_H}$ is a sink place labeled by v, indicating the overall violated state of NN^i, and*
- *M is the current marking of the CPN model $\mathcal{N}_H$.*

The initial marking of the CPN model of a norm net instance is indicated by the initial normative state of the norm net instance, from which the normative state of the norm net instance will transit with the occurrence of events and accordingly the marking of the CPN model. However, not all markings in a CPN model have a corresponding normative state for the norm net instance, as one event may simultaneously trigger several transitions in the CPN model (the transitions with the same prefix). Only the two markings, the one before and the one after the event, will have a corresponding normative state in the norm net instance. To this end, given the occurrence of an event, we need to identify all the enabled steps (see Definition 2.3) that can occur in the corresponding CPN model, as shown in Definition 5.9.

Definition 5.9 (Enabled steps) *Given an external event $e = (ag, \alpha)$, the set of all enabled steps in a hierarchical CPN model $\mathcal{N}_H$ with the current marking M is $EStep(e, \mathcal{N}_H, M) = \{(t_1, b_1), (t_2, b_2), ..., (t_n, b_n)\}$ such that*

- $t_i \in T_{\mathcal{N}_H} - T_{sub\mathcal{N}_H}, 1 \leq i \leq n : f_{\text{ACTION}}(t_i) = \alpha$, *indicating that* t_i *is a non-substitution transition in the hierarchical CPN model* $\mathcal{N}_H$ *and is labeled as* α *(the name of the action),*

- $\exists(p, t_i) \in A_{\mathcal{N}_H}$: "*ag*" $\in M(p)$, *indicating that there is an input place of* t_i *which contains a token carrying the value "ag" (the name of the agent) in the current marking* M*,*

- $\forall(p, t_i) \in A_{\mathcal{N}_H} : E(p, t_i)\langle b_i \rangle \leq M(p)$, *indicating that a transition instance of* t_i *is enabled in the current marking* M *with a binding* b_i*, and*

The first two requirements indicate that when an agent *ag* performs an action α, characterized as an external event, we will first search in the hierarchical CPN model of a norm net instance for all the non-substitution transitions named α, and then check for each of these transitions whether there is a token carrying the value *ag* in one of its input places. As a result, we will get a set of transitions that correspond to the occurrence of the external event. For example, given that an external event (*Bob*, *send_email*) occurs in the physical world, we will first look for all the non-substitution transitions labeled *send_email* in the hierarchical CPN model of the norm net instance, and then check for each of these transitions whether one of its input places contains a token with the value *Bob*. The last two requirements in the definition ensure that these qualified transitions are enabled to fire according to the rules of enabled transitions in CPNs.

Given the set of enabled steps *EStep*$(e, \mathcal{N}_H, M)$, $M \xrightarrow{e} M'$ refers to a set of occurrence sequences. For example, an occurrence sequence could be $M \xrightarrow{(t_1,b_1)} M_1 \xrightarrow{(t_2,b_2)} M_2 ... \xrightarrow{(t_m,b_m)} M'$ where M, M' are two markings in the CPN model and $(t_i, b_i) \in$ *EStep*$(e, \mathcal{N}_H, M)$. Those M_i in the sequence are intermediate markings which do not have corresponding normative states in norm net stances. It should be noticed that there are no order constraints on (t_i, b_i) in the sequence. As a result, M' is reachable from M with respect to the originating norm net if and only if $\mathscr{R}_{CPN}(M, M')$ and there is a permutation Pt on the sequence of (t_i, b_i) such that $\forall pt \in Pt, \exists e, pt = EStep(e, \mathcal{N}_H, M)$.

Definition 5.9 shows that the normative states and their transitions in norm net instances are respectively mapped to the markings and enabled steps in CPNs, which means that the dynamics of NNs is fully captured by CPNs. The techniques that are invented originally for CPNs are now available for NNs, e.g., state space analysis techniques.

5.3 Properties of Norm Nets

In this section, we give an analysis of the properties of NNs in terms of the types of norms and normative relations that can be represented in the framework proposed by [89].

5.3.1 Properties of Single Norms

The properties of single norms hold for every norm instance.

(1) Violation

In the CPN model of a regulative norm, we explicitly indicate the satisfied and violated state of corresponding norm stances. For an obligation instance, the CPN model indicates that if the target occurs before the deadline, the instance is satisfied, otherwise it is violated. For a prohibition instance, if the target occurs before the deadline, the instance is violated, otherwise it is satisfied. However, it is possible that the target and the deadline occur simultaneously. In CPNs, concurrent behaviors are modeled by the interleaving semantics, i.e., the next markings are obtained by considering all the possible firings of all transitions, one after the other, starting from the same state, and leads to n next markings when n transitions are enabled in parallel. However, for the situation of simultaneous occurrence of the target and the deadline of a norm instance, there will be a race condition between the corresponding transitions for certain markings, i.e., for markings having several enabled transitions and for which the firing of some of these transitions makes some other transitions no longer firable.

Specifically, in the CPN model of a regulative norm instance, if the precondition is satisfied, the compliance evaluation with respect to the target and the deadline is enabled. That is, at this time, if the target and the deadline are satisfied simultaneously, we are confronted with a race condition between the firing of the target and the firing of the deadline. Suppose that the target fires first, then the compliance evaluation of the norm instance results into a satisfied state in case that the norm is an obligation, while the deadline will be disabled. On the other hand, if the deadline fires first, then the compliance evaluation of the norm instance results into a violated state in case that the norm is an obligation, while the target will be disabled.

It can be seen that we get two inconsistent results from the two firing choices. To this end, we make two assumptions (1) it is not allowed to have a norm whose target and deadline correspond to the same events, (2) the occurrence of two distinct events is always in a sequential order, i.e., they will never occur simultaneously. As such, the race condition between the target and the deadline of a norm can be avoided in the operations of CPNs.

(2) Non-preemptive

- Definition: (i) for every obligation instance, the events representing the target occur before the events representing the precondition, does not lead the instance to be satisfied; (ii) for every prohibition instance, the events representing the target occur before events representing the precondition, does not lead the instance to a violated state.

- Example: (n^r_{31}) If a student borrows a book from the library, he or she should return the book within 1 month; (n^r_{32}) If a student enters the library, he or she should not talk until he or she leaves the library.

- Representation in NNs:

  – n^r_{31} = (O, (*student*, *return_book*), (*Timeline*, *pass_1m*), (*student*, *borrow_book*)), and

  – n^r_{32} = (F, (*student*, *talk*), (*student*, *leave_library*), (*student*, *enter_library*)).

Based on the normative state transitions of obligations and prohibitions we have presented in Section 5.1.5. The normative state transition diagrams of the two norms n^r_{31} and n^r_{32} are shown in Figure 5.14. It can be seen that only when the events (*student*, *borrow*) and (*student*, *enter_library*) respectively representing the precondition in n^r_{31} and n^r_{32} occur, the instances of the obligation n^r_{31} and the prohibition n^r_{32} can be activated. Only after the norm instances are activated, the satisfied and violated can be reached. In this sense, when (*student*, *return_book*) occurs before (*student*, *borrow*), the satisfied state cannot be reached. Similarly, when (*student*, *talk*) occurs before (*student*, *enter_library*), the violate state cannot be reached.

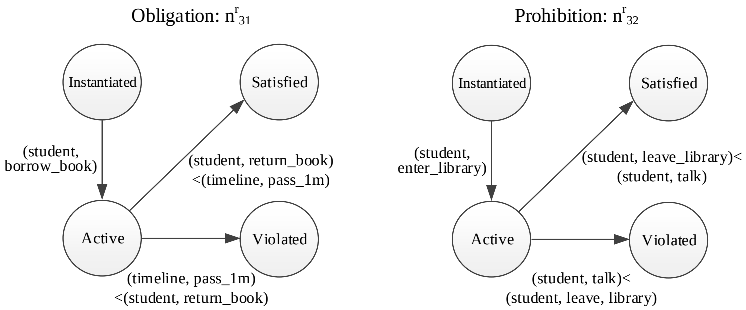


Figure 5.14: *State transition diagrams of the two norms* n^r_{31} *and* n^r_{32}*.*

(3) Non-persistent

In NNs, regulative norms are non-persistent in the sense that *for every norm instance, as soon as the deadline of the norm instance occurs the norm expires, i.e., the norm instance has no regulation effect any more.*

The obligation to fulfill the target itself may persist after the deadline but the deadline signals whether a violation has occurred. Taking the obligation n^r_{31} as an example, if the student does not return the book on time, even after a month, the obligation to return the book should still be fulfilled though returning later would not invalid the violation of "not returning book on time". While as another example, students should submit their answer sheets within 5 minutes after the teacher signals the end of the exam, otherwise the teacher should revoke the student's qualification of the exam. In this case, if a student does not submit the answer sheet in time, there is no sense to fulfill the obligation after the deadline occurs.

(4) Achievement and Maintenance

In NNs, a regulative norm has either an achievement or a maintenance nature, in the sense that

- *obligations are of achievement nature: for any obligation to be satisfied, after the precondition occurs the target must occur at least once before the occurrence of the deadline.*
- *prohibitions are of maintenance nature: for any prohibition to be satisfied, after the precondition occurs the target should not happen during all instants before the deadline occurs.*

For example, the achievement property of the obligation n_{31}^r can be derived from the fact that, as shown in Figure 5.14, after the occurrence of the event (*student*, *borrow_book*) representing the precondition, only when the event (*student*, *return_book*) representing the target occurs before the event (*Timeline*, *pass_1m*) representing the deadline, a satisfied state can be achieved. Similarly, the maintenance property of the prohibition n_{32}^r can be derived from the fact that, as shown in Figure 5.14, after the occurrence of the event (*student*, *enter_library*) representing the precondition, only when the event (*student*, *leave_library*) representing the deadline occurs before the event (*student*, *talk*) representing the target, a compliant state can be maintained.

5.3.2 Properties of Norm Relations

The properties of norm relations discussed here are the ones that can be represented by NNs.

(1) Contrary to Duty

- Definition: when a regulative norm cannot be fulfilled, there can be a sub-ideal solution by imposing another regulative norm.
- Example: (n_{32}^r); if the student talks in the library before he or she leaves the library, the student should lower his or her voice (n_{33}^r).
- Representation in NNs: OE(n_{32}^r, n_{33}^r) where
  - n_{33}^r=(O, (*student*, *lower_voice*), (*student*, *leave_library*), λ).

In NNs, we refer to the two component regulative norms (or regulative norm nets) connected by an OE operator as the origin and the reparation (or sanction). Only when the origin is violated, the reparation can be imposed. With the OE operator, we differentiate between an idea status and a sub-ideal status by the two compliance evaluation states C and C^* in Table 5.2. In this way, NNs are able to capture the contrary-to-duty relation in normative systems.

(2) Exceptions

- Definition: A regulative norm does not apply when there are certain exception conditions and at the same time the exception conditions hold.
- Example: (n^r_{32}); except that the librarian gives permission to the student before he or she talks in the library (n^r_{34}).
- Representation in NNs: $OR(n^r_{32}, n^r_{34})$ where
  - n^r_{34}=(O, (*librarian*, *give_permission*), (*student*, *talk*), λ).

An exception, to a norm (or a set of norms), indicates that there is another alternative to be satisfied. In NNs, an OR relation indicates a choice between two alternatives. It means that as long as one of the alternatives is satisfied, it does not matter whether or not the other alternative is fulfilled. In this way, NNs are able to capture an exception relation in the sense that the exception can be modeled as an alternative to the fulfillment of another norm. However, it has to be noticed that one needs to take a combined view on the two component regulative norms when using an OR operator to represent their exception relations.

(3) Sanction and Reparation

- Definition: the violation of a regulative norm (or regulative norm net) leads to the activation of another regulative norm (or regulative norm net); the activation of the latter is seen as the sanction of violating the former and the fulfillment of the latter can repair the violation of the former.
- Example: (n^r_{32}); otherwise the student must make a formal apology before he or she leaves the library (n^r_{35}).
- Representation in NNs: $OE(n^r_{32}, n^r_{35})$ where
  - n^r_{35}=(O, (*student*, *make_apology*), (*student*, *leave_library*), λ).

A sanction to a norm (or a set of norms) indicates that some specified measures have to be imposed when the norm (or the set of norms) is violated. A reparation to a norm (or a set of norms) indicates that it is possible to remedy the violation of the norm (or the set of norms) by means of some specified measures. The representation of such relations requires a norm specification to be able to differentiate between two component norms (or norm sets), and a mechanism to activate one component norm (or norm set) in case of the violation of the other. In NNs, with an OE operator, two regulative norms (or regulative norm nets) are differentiated as the origin and the reparation (or sanction). When the origin is violated, the reparation (or sanction) is activated, and the fulfillment of the latter can lead to a satisfaction as a whole. In this way, NNs are able to capture the sanction and reparation relations, as well as chains of sanction and reparation, in normative systems.

5.4 Compliance Checking of Agent Behavior

To ensure the achievement of organizational compliance, a main concern is whether the behavior of the participants are complying with the compliance rules that are relevant to the regulation of organizational behavior. That is, we have to check whether the actions of the agents are in accordance with the normative constraints specified in the normative structure of the organization model.

As described in the previous chapter, agents participate in an organization by enacting one or more specified roles of the organizations and they interact with each other to pursue the global objectives of the organization. The interactions between the agents can be reflected from the fact that their actions have impacts on each other. For example, from the normative perspective, an agent might be subject to some obligations when other agents take particular actions. To indicate the interactions of agents, we give the definition of an interaction plan as follows.

Definition 5.10 (Interaction plan) *An interaction plan* $IP = (e_1, \ldots, e_n)$ *where* e_i *is an instance of an event from the set of external events* Ev, $1 \leq i \leq n$.

An interaction plan presents a temporal order of *who does what* in the physical world by means of external events (agent-action pairs). Here we use the form of a sequence instead of more complex structures since the interaction plans are perceived as the observation of the occurrences in the physical world. Nevertheless, a complex structure of agent interactions can also be applied by abstracting all the possible sequences.

With respect to an interaction plan, the occurrence of the sequence of events in the plan are the triggers to the changes of normative states of regulative norms through the interpretation provided by constitutive norms. As illustrated before, we use markings of CPNs to represent normative states of NNs, by which the transition of normative states can be captured by the transition of marking with respect to the concurrence of events. In this sense, we can make use of the CPN mapping of NNs to check norm compliance of agent behavior.

As presented in Section 5.1.5, we use the Compliance Evaluation State (CES) as a projection of the normative state of norm net instances to indicate the compliance status of the norm net instances. In the CPN model of a norm net instance, there is a sink place labeled v and colored violated. When there is no token in that place, the *CES* of the norm net instance is evaluated to be compliant, otherwise it is evaluated to be violated. In this sense, to know the impact of the occurrence of an event on the *CES* of a norm net instance, we can check whether the number of tokens in the violated place of the CPN model is changed. That is, given the occurrence of an external event, the transitions in the CPN model of the norm net instance will fire according to the enabled steps obtained from Definition 5.9, which changes the marking of the CPN model accordingly. By checking the marking of the violated place (indicated by the place named v in Figure 5.8, 5.9 and 5.10), we are able to determine whether the corresponding norm net instance is compliant or violated, as described below.

Definition 5.11 (Compliance evaluation) *Given the CPN mapping* $\Theta(NN^i) = \langle(\mathcal{N}_H, P_s, P_v, P_{vr}, p_{so}, p_{vo}), M\rangle$ *of a norm net instance* NN^i*, the compliance evaluation state CES of* NN^i *is either*

- ***compliant*** *if* $M(p_{vo}) = \emptyset$*, or*
- ***violated*** *otherwise.*

However, when a norm net instance is in the compliant state, it does not mean that every component norm (or norm net) in the norm net instance is in the compliant state. For example, there still might be violated norms as a branch of an OR relation while the other branch is in the compliant state. In the case of an OE relation, even if a violation has been repaired (by fulfilling the sanction), the CPN model still records the violation of the norm (indicated by the place labeled vr in Figure 5.10). This is important for the verification of full compliance (or ideal compliance) and is captured as below.

Definition 5.12 (Reparation evaluation) *Given the CPN mapping* $\Theta(NN^i) = \langle\ (\mathcal{N}_H, P_s, P_v, P_{vr}, p_{so}, p_{vo}), M\rangle$ *of a norm net instance* NN^i*, the compliance evaluation state CES of* NN^i *is*

- ***sub-ideal compliant*** *if* $M(p_{vo}) = \emptyset$ *and* $\exists p \in P_{vr} : M(p) \neq \emptyset$*.*

There can be multiple repairable violation places in the CPN model of a norm net instance since it is possible to have more than one OE relation in the norm net specification. As long as there is a token in any of the places labeled vr in the CPN model of the norm net instance, the repair evaluation of the norm net instance is considered to be sub-ideal compliant.

We have shown that using the CPN model of a norm net instance, the compliance evaluation state of the norm net instance can be obtained by checking the marking of the corresponding CPN model. Therefore, given an interaction plan, we can query the state changes of the norm net instance by checking the corresponding markings of the CPN model after the occurrence of each event in the plan, as captured in Algorithm 2. A detailed description of this algorithm is presented in Appendix A.7.

Algorithm 2 Compliance Query: CLQ

Require: (NN^i, IP) ▷ A norm net instance and an interaction plan
Ensure: CE ▷ Compliance evaluation results

▷ Obtain the new marking of a CPN model $\mathcal{N}$ given the occurrence of an enabled step Y with the current marking M
function UPDATESTATE($Y, M, \mathcal{N}$)
    ▷ The body of this function is presented in Appendix A.7
return M'
end function

▷ Evaluate the compliance evaluation state given two normative states
function COMPLIANCEEVA(M, M', P_{vr}, p_{vo})
    if $M'(p_{vo}) - -M(p_{vo}) \neq \emptyset$ **then**
        $ce \leftarrow$ *violated*
    else
        $ce \leftarrow$ *compliant*
    end if
    if $ce =$ *compliant* **then**
        for all $p \in P_{vr}$ **do**
            if $M'(p) - -M(p) \neq \emptyset$ **then**
                $ce \leftarrow$ *sub-ideal compliant*
            end if
        end for
    end if
return ce
end function

▷ (**Step 1**) Obtain the CPN mapping of the norm net instance NN^i
$\langle(\mathcal{N}, P_s, P_v, P_{vr}, p_{so}, p_{vo}), M\rangle \leftarrow \Theta(NN^i)$

▷ (**Step 2**) Obtain the changes of the compliance evaluation state of NN^i with respect to the interaction plan IP
for $j \leftarrow 1$ to $|IP|$ **do**
    ▷ (**Step 2.1**) Obtain the new normative state of NN^i given the occurrence of the event $IP[j]$
    $M' \leftarrow$ UpdateState(EStep($IP[j], \mathcal{N}, M), M, \mathcal{N}$)
    ▷ (**Step 2.2**) Evaluate the compliance status of NN^i with respect to the occurrence of the event $IP[j]$
    $(CE[j]) \leftarrow$ ComplianceEva(M, M', P_{vr}, p_{vo})
    $M \leftarrow M'$
end for

Given an interaction plan IP, the CPN model of NN^i is able to execute accordingly. Each event in the interaction plan IP might cause a change to the marking of the CPN model and thus the compliance status of NN^i. From the compliance query results, we know the influence of the interaction plan on the compliance status of NN^i. That is, we can determine whether a specific interaction process is complying with a set of norms. If and only if the CES is *compliant* at all steps, the given IP is considered as a full-compliant plan with respect to NN^i, as formalized in Definition 5.13.

Definition 5.13 (Full-compliant interaction plan) *An interaction plan $IP = (e_1, e_2, ..., e_n)$ is a full-compliant plan with respect to a norm net instance NN^i iff* $\forall(CES[j]) \in CLQ(NN^i, IP) : CES[j] = compliant,\ 1 \leq j \leq n.$

Full compliance is important for regulated environments because the reparation of a violation may still cause a heavy loss to the actors. Therefore, from the organizational perspective, it is necessary to check whether an interaction process can achieve a full compliance against all the relevant norms.

To check whether a given interaction plan is full-compliant with respect to the CPN model of a norm net, the main calculation lies in the firing of the relevant transitions for each event in the sequence. Given an event, there are three possibilities for each transition in the CPN model: (1) the transition is not relevant, (2) the transition is relevant but not enabled, and (3) the transition is enabled to fire, i.e., consumes the tokens in its input places and produces a token in each of its output place. In this sense, we can transform the problem of full compliance checking of a given interaction plan to the membership problems in Petri Nets which is decidable [98]. The complexity of the full compliance checking can be done in efficient time (polynomial in the length of the interaction plan and the number of transitions in the CPN model of a norm net). However, in order to query all the possible full-compliant paths in the CPN model of a norm net which is undecidable, we are currently researching pruning mechanisms to reduce the search space by eliminating meaningless states.

5.5 Discussion

In Section 2.5 and 2.7, we have summarized the literature on norms and compliance checking. There are two lines of works that are closely related to NNs. One is based on the notion of alignment (e.g., [172, 3, 137]) and the other is based on the notion of commitments (e.g., [194, 221, 196]). In this section, we discuss the commonalities and differences between these works and ours.

5.5.1 Representation of compliance rules

To facilitate automated compliance checking, it is important to represent compliance rules in formalized specifications, providing a precise and unambiguous description of the expected behavior. For this purpose, the alignment-based approaches have used Petri nets (e.g., [172]) and declarative process models (e.g.,

[138]) as a representation of compliance rules. To bridge the gap between informal compliance rules and a precise specification, an interactive question and answer (Q/A) based approach was proposed along with a repository of compliance patterns [174, 173]. In the commitment-based approaches (e.g., [196]), a rich set of constructs and types of norms are provided. Each norm contains a subject, an object, a context, an antecedent and a consequent while a norm can be of types commitment, authorization, prohibition, sanction, power. Moreover, a unified life cycle is specified for all types of norms. By applying natural language processing (NLP) and machine learning (ML) techniques, an approach was proposed to extract these different types of norms from business contracts [73].

While in our approach, we model compliance rules in the Normative Structures through the formalism of NNs, which feature building blocks interpreting high-level concepts and is grounded by counts-as conditionals and deontic logic. The building blocks in Normative Structures include roles, conditions, deadlines, obligations, prohibitions, reparations/sanctions and norm inter-relationship. These building blocks serve a similar function as the compliance rule patterns and interactive Q/A based compliance rule composing adopted in the alignment-based approaches, though via different methodologies. The former provides a formal language to describe compliance rules while the latter provides a tool for assisting compliance rule composition in lower-level languages. As to the commitment-based approaches, both their models and our models use similar modeling components and norm states. For example, the precondition and target in NNs respectively corresponds to the antecedent and consequent in commitment-based norms. Except for the state of pending, an alignment may be found between the norm state changes in commitments-based approaches and NNs. With respect to the formalization of compliance rules, it would be an useful feature if the NLP and ML based norm extraction proposed for commitment-based approaches can be adapted for NNs.

As a high-level concept, the interrelations between compliance rules are often neglected or assumed to be conjunction, as the case in alignment-based and commitment-based approaches. In NNs, we explicitly capture the interrelations between compliance rules (e.g., disjunction and reparation/sanction) to facilitate/guide domain experts in expressing sets of compliance rules. The explicit representation of such interrelations also enables a modular way of formalizing sets of compliance rules.

5.5.2 Evaluation of Compliance

Alignment-based approaches and NNs both rely on the records of occurrences in the reality, i.e, backward compliance checking. In particular, alignment-based approaches advocate the use of enriched logs which contain detailed information on the occurrence of events such as time of occurrence, resource usage, etc. With the rich information, a detailed evaluation of compliance is realized in the alignment-based approaches, which is interpreted as degrees of conformance and provides a basis for finding optimal solutions to correcting the misbehavior [138]. In NNs, the degree of compliance is captured in three levels, namely compliance, sub-ideal compliance and non-compliance/violation. As a direction for future research, we plan to empower NNs with detailed normative evaluation methods and corresponding techniques to find optimal ways of avoiding/correcting violations.

Besides norm compliance, the Preference Structure to be discussed in Chapter 10, provides another dimension of information regarding evaluating consequences of the occurrences in the reality, which can be seen as compliance to personal rules and is an important aspect in agent-based systems. Such a perspective has also been considered in the commitment-based approaches by allowing principals/participants to choose their own policies [196].

5.6 Conclusions

In this chapter, we introduced a normative language Norm Nets (NNs) for the construction of the normative structure of OperA+. Based on the notion of norms, NNs provide a formal representation for compliance rules. By incorporating both constitutive and regulative norms, NNs are able to provide a link between the occurrences in the physical world and the institutional consequences perceived by the organizations. Moreover, NNs provide a modular way of representing the compliance relations between norms, which facilitates the reusability of normative design and more importantly enables an analysis of normative properties in a more sophisticated way.

Through a mapping from NNs to Colored Petri Nets, the operational semantics of NNs are obtained, which facilitates the compliance evaluation of agent interactions. The approach is able to provide information on the compliance consequence of both individual agents' behavior and the collective behavior perceived by the organization. That is, from an organizational point of view, we can determine whether the whole normative system is violated or not by checking the existence of tokens in some featured places in the CPN models of NNs. From an individual perspective, we can determine which agents violate the norms by checking the distribution of the tokens in the CPN models. In this way, not only can we determine whether the behavior of the agents is in accordance with the norms specified by the organization, but also provide a potential approach of abstracting interaction processes that are norm compliant. For example, we can search for full-complaint occurrence sequences in the state space of the CPN model of a given norm net instance. This creates priori knowledge for process designers and facilitate process implementation in regulated business environments.

Chapter 6

Normative structure:
Conflicts Detection and Contextualization

This chapter has the following contributions:

- introducing the concept of governance scopes of institutions and providing a formalization of this concept,
- presenting a formalization of norm conflicts and providing a computational mechanism of detecting such conflicts,
- presenting a model for contextualization of normative structure.

In the previous chapter, we have presented a normative language Norm Nets (NNs) for the construction of normative structures in OperA+ and illustrated the mechanism that can be used for checking whether agent interactions are complying with the norms specified by a normative structure. The scope of study was a single institution[1] which prescribes a set of norms to regulate agent interactions.

Organizational regulation is rarely self-sufficient to ensure the achievement of organizational objectives, especially in a multi-organizational interaction setting, but depends on various (external) regulation sources such as legal regulations, business agreements, and industrial best practices, etc. For example, in virtual organizations (VOs) [163] where various parties are involved to achieve individual

Parts of this chapter have been published. Section 6.1, 6.2 and 6.3 are based on our contribution to the 13th International Workshop on Coordination, Organisations, Institutions and Norms @ AAMAS, pp. 136-154, 2013 [141]; Section 6.4 is based on our contribution to the 13th International Workshop on Coordination, Organizations, Institutions, and Norms (COIN), pp. 141-157, 2012 [117].

[1] We follow the definition provided in [165] that an institution is a set of prescriptions that humans use to organize all forms of repetitive and structured interactions.

and global objectives, multiple institutions may be employed to cover different aspects of regulating the behavior of the participants to ensure that the VO's objectives can be achieved in a desired way. Given the diversity of regulation sources and possibly conflicting interests, it is likely that the institutions are not consistent in regulating the behavior of the participants. This raises a question on an important aspect of organizational compliance, i.e., how to deal with the combination of multiple sources of compliance requirements with respect to the issue of consistency. We use the concept of institution to represent a set of compliance rules imposed by one regulation source.

In the previous chapter, we have shown how a set of compliance rules can be formalized by means of a set of norms expressed in a norm net. In the setting of multi-institutions, it is not only the norms that make up an institution inherently serve to indicate its applicability, but also the variables, in terms of which those norms are expressed, are typically intended to be restricted to specific, meaningful ranges within the domain being modeled. That is, the regulation of an institution also needs to characterize the situations (being particular combinations of contextual information such as time, location, etc.) in which a given institution has competence. To this end, we introduce the concept of *governance scope* and investigate the coordination of multiple institutions. Each institution specifies a set of norms covering a specific aspect of the problem domain with a governance scope defining its remit. Together, they govern the participants and reflect the objectives of the organization. With the participants' behavior being simultaneously regulated by more than one institution, normative conflicts can appear, i.e., the normative specification of different institutions may be inconsistent. In order to detect such inconsistencies, we present a computational mechanism by means of the CPN models of NNs.

In practice, norms from various institutions are usually not specified at a single level of abstraction [117]. At a high abstraction level, norms tend to range over many different situations and to require little maintenance over time. At a lower operational level, the abstract norms are related to the concrete events/concepts that occur in the system. Proposed by Searle [188], counts-as statements add institutional meaning to real-world facts, which have been studied by researchers to connect abstract concepts to more concrete ones (e.g., [9], [91]). Such connections are a process of refining norms from abstract values to operational details and the refinement process is determined by the environmental conditions (contexts) in which the norms are applied. Therefore, it is necessary to reflect how different contexts are integrated with the refinement of norms. To this end, we present a context refinement process by which NNs can be elaborated from abstract concepts to concrete ones with respect to differing application environments.

This chapter is organized as follows. Section 6.1 introduces the model of contextualized institutions. Section 6.2 presents how multiple institutions are coordinated. Section 6.3 introduces the concept of norm conflict and illustrates the mechanism of conflict detection. Section 6.4 illustrates how norms are refined from abstract to concrete concepts. Section 6.5 concludes this chapter.

6.1 Contextualized Institutions

As described in [141], we regard the process of designing an institution as not only the definition of a set of norms but also the characterization of its governance scope, i.e., what kinds of situations are under control of the institution, since this is what gives the institution its 'footprint'. That is, the same set of norms with different governance scopes results in different *contextualized institutions*. Figure 6.1 shows the framework of the governance model in a multi-institution setting, consisting of three parts:

1. *contextualized external events* occurred in the physical world, each of which has associated contextual information that characterizes the situation where the event occurs,

2. *institutions* comprising sets of norms, in which *constitutive norms* translate external events into institutional events, and *regulative norms* (obligations and prohibitions) bring about the changes of normative states according to the generation of institutional events, and

3. *governance scopes* that delineate the control boundary of institutions through a set of contextual dimensions.

With governance scopes, contextualized institutions are built, which facilitates the identification of applicable institutions for a given event. In the remainder of this section, we explain each of these components in more detail.

6.1.1 Contextualized External Event

External events are the occurrences of agent actions in the physical world. Except the basic elements defined in an external event (agent and action), other contextual information, such as when and where the event occurs, can also be included to refine the occurrence of the event. Therefore, we extend the definition of an external event by adding a set of contextual elements, which permits us to correlate events with the customized contextual dimensions of governance scopes.

Definition 6.1 (Contextualized external event) *A contextualized external event $\widehat{e}$ is a tuple (e, ϑ) where*

- *e is an external event as described in Definition 5.1,*

- *$\vartheta \in \prod_i V_i$, $V_i \in D$ characterizes the situation where the event occurs, with respect to a set of contextual dimensions (see Definition 4.6).*

For example, given two dimensions $D = \{Location, Time\}$, an event could be $\widehat{e} = ((Alice, have\_ lunch), (McDonald's, 1pm))$, indicating that Alice performs the action of having lunch at the time of 12pm and at the location of McDonald's.

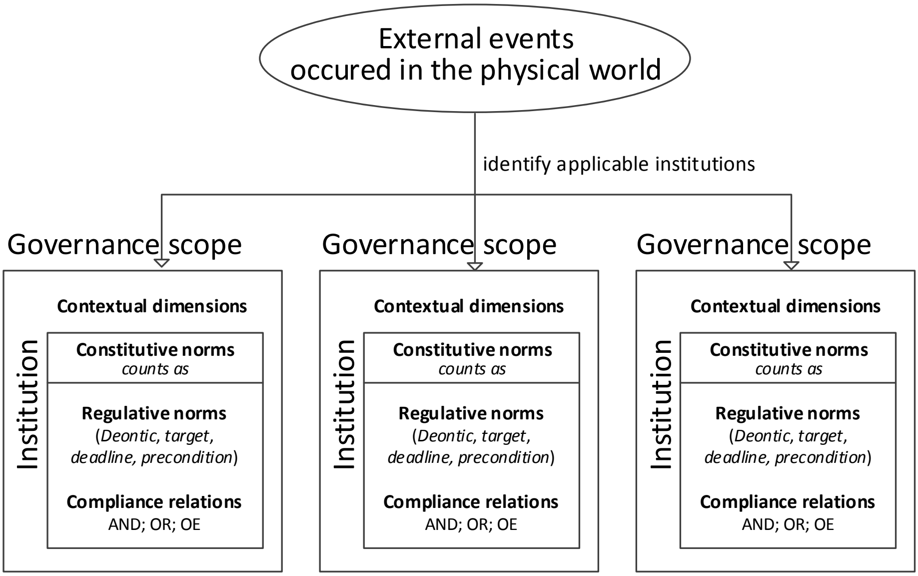


Figure 6.1: *Governance model in a multi-institution setting.*

6.1.2 Governance Scope

Governance scopes are used to delineate the control boundary of institutions, determining which situations are under their control. To capture the governance scope of an institution, we adopt the context model presented in Definition 4.6 in Chapter 4. Correspondingly, we characterize a governance scope as a set of contextual dimensions. Different contextual dimensions indicate different ways of establishing the governance scope of an institution. For example, an institution can specify its governance scope by defining a contextual dimension of *Individuality*, indicating that as long as the entities evolved in an event belongs to a set of individuals, the institution has the right to govern the behaviour. Similarly for *Location*, an institution can indicate that as long as the observed location of an event belongs to a set of locations, the behaviour is in the governance scope of the institution. Definition 6.2 formalizes *Governance Scope.*

Definition 6.2 (Governance scope) *A governance scope $gs \subseteq \prod_i V_i$, $V_i \in D$, relating to a set of contextual dimensions.*

It can be seen that gs specifies a multi-dimensional space by assigning each contextual dimension a set of values. In practice, a governance scope might have no constraint on a particular contextual dimension. In this case, a value of complete set denoted as U is assigned to that dimension and we consider the governance scope covers the whole value set of that contextual dimension. For example, a governance scope could be $\{Italy, France\} \times U_{time}$.

6.1.3 Institution

As stated before, institutions are not only a set of norms but also characterized by governance scopes which reflect their control boundaries. To this end, we extend the model of NNs such that the control boundary of an institution can be captured as well.

Definition 6.3 (Contextualized norm net) *A contextualized norm net is defined as a tuple* $\widehat{NN} = (gs, RNN, CNN)$*, where*

- gs *is a governance scope,*
- RNN *is a regulative norm net (see Definition 5.6),*
- CNN *is a constitutive norm net (see Definition 5.6).*

Note that gs might incorporate contextual dimensions such as individuality and activity which are different from the set of agents Ag and the set of actions $\mathcal{A}$ in the specification of CNN. The elements in gs are used to delineate the control boundary of the institution modeled by the norm net $\widehat{NN}$ while the elements in Ag and $\mathcal{A}$ are used to specify all the agents and actions that are recognized by the institution.

With governance scope, a norm net specifies all the situations that are under the regulation competence of an institution. Two institutions might impose the same set of norms while have different governance scopes. Given an external event with extended contextual information, we first use the values of the contextual dimensions to determine whether the event falls in the governance scope of an institution. If so, then the event is (partially) translated into institutional events since some of the contextual information might not be relevant for the regulative norms and are only needed for the determination of governance scopes.

6.2 Collective Institutions

Institutions are designed originally for their own regulation purposes and thus have specific governance scopes. As long as the institutions are internally consistent, they can successfully operate independently. In settings (e.g., virtual organizations) where interactions are governed by multiple institutions, however, mutually exclusive norms might be imposed by the institutions with overlapping governance scope. For example, when a Dutch citizen applies for a visa to the USA, several institutions might be triggered, e.g., US embassy, Dutch government, and a conflict could exist between information requirements from the US embassy and privacy policies from the Dutch government. Therefore, to know the consistency of multiple institutions, we first need to understand how institutions cooperatively work together in regulating participants' behavior.

Figure 6.2 shows an example of how a set of institutions, captured by a set of contextualized norm net instances, evolve with a sequence of events occurred in a virtual organization. At the initial state, all the institutions are instantiated. When an event occurs, only those institutions whose governance scope covers that event will be activated. We can see that at different time instants, there are different sets of institutions activated by the same event. That is, the contextual information of an event simultaneously maps to the governance scope of some institutions. To represent these simultaneously activated institutions, we introduce the concept of *Collective Institution Set* in Definition 6.4.

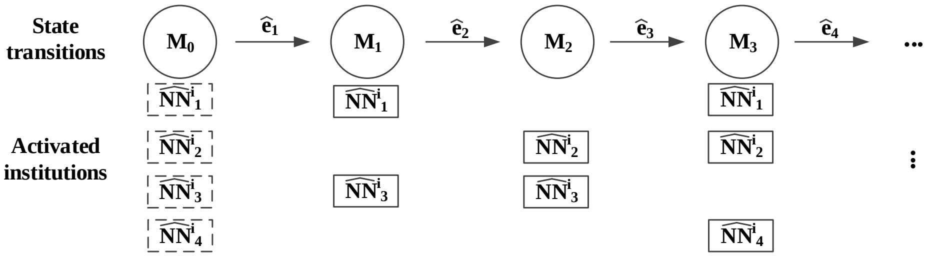


Figure 6.2: *Cooperative regulation of multiple institutions.*

Definition 6.4 (Collective institution set) *In an organization governed by a set of institutions captured by a set of norm net instances* $\{\widehat{NN}^i_1, ..., \widehat{NN}^i_m\}$, $\widehat{NN}_j = (gs_j, RNN_j, CNN_j)$, $1 \leq j \leq m$, *given a contextulaized external event* $\widehat{e}$ *occurring at time instant* k, *a collective institution set is defined as* $\mathcal{C}_k = \{\widehat{NN}^i_j | \vartheta_{\widehat{e}} \in gs_{\widehat{NN}^i_j}, 1 \leq j \leq m\}$.

At any time instant k, the set of all institutions whose governance scope covers the contextual information of the event that occurs at time k is called a collective institution set, indicating all the activated institutions given the occurrence of an event. The governance scopes of these institutions overlap with each other and thus they all have governance competence on the same event.

In a collective institution set, the overlap relation between governance scopes is indicated by the same external event covered by the set of institutions (represented by the set of norm net instances) in the collective institution set. The overlap relation is determined by the values of each contextual dimension with respect to the governance scope of the institutions. To represent the overlap between the governance scopes of different institutions, we introduce the concept of *Governance Overlap*.

Definition 6.5 (Governance overlap) *Given two governance scopes gs and gs', the governance overlap Ω between gs and gs' is defined as $\Omega(gs, gs') = gs \cap gs'$.*

If $\Omega(gs, gs') \neq \varnothing$, we say gs and gs' have a non-empty overlap. In particularly, a collective institution set $\mathcal{C}_k = \{\widehat{NN}^i_1, \ldots, \widehat{NN}^i_m\}$ implies that $\forall \widehat{NN}^i_j, \widehat{NN}^i_l \in \mathcal{C}_k, \Omega(gs_{\widehat{NN}^i_j}, gs_{\widehat{NN}^i_l}) \neq \varnothing$ as they all have governance of a specific event. This means that the governance scopes of all the institutions in a collective institution set have a non-empty overlap.

Given the cooperative regulation of multiple institutions, we now turn to the investigation of norm conflicts in institutions.

6.3 Norm Conflicts

The regulation of an institution is realized through the specification of a set of norms on the actions of the participating agents. This set of norms should be consistent, otherwise we cannot reach an agreement on the compliance of the occurrence of certain events. Therefore, it is essential to check the consistency of the set of norms, which is often known as norm conflict detection in the literature (e.g., [207], [79]). A conflict occurs between an obligation and a prohibition when they constrain on the same agent behavior and have an overlapped activation period (cf. [133]). That is, if some actions of an agent are obliged and forbidden at the same time, a conflict arises. From this definition, we differentiate between two types of norm conflicts. First, *weak conflicts*: the activation period of the prohibition does not cover the whole activation period of the obligation. In this sense, a weak conflict is a conflict that can be avoided when the event, being constrained by the two norms, occurs during the time period where the obligation is activated while the prohibition is not. In this way, both norms can be satisfied. Second, *strong conflicts*: the activation period of the prohibition covers the whole activation period of the obligation. A strong conflict is a conflict that cannot be avoided. That is, whenever the event occurs, one of the norms will be violated.

In the setting of (virtual) organizations where multiple institutions are employed, norm conflicts may concern norms within a single institution as well as from different institutions. In the case of norms coming from two (or more) institutions, norm conflicts may occur only when the institutions have an overlapped governance scope over the event being constrained by the norms from both sources. That is, the institutions have to be in the same collective institution set with respect to the occurrence of the event. Moreover, a similar criterion lies in both cases of a single institution and multiple institutions, i.e., there are contradictory compliance judgments (satisfied or violated) with respect to the occurrence of an event. That is, with the event occurring, there are two (or more) norms in which the normative state of one norm is evaluated to be satisfied while that of the other norm is evaluated to be violated, which means that the institution(s) cannot reach an agreement on whether the event is complying with the norms specified by the institution(s). Remember that when talking about states of norms, we always refer to the instances of norms (or norm nets).

Furthermore, to determine whether a norm conflict exists, there is another criterion that has to be considered, i.e., the compliance relation between norms. As for multiple institutions in a collective institutions set, we currently do not consider ranking in their regulation power, and the compliance relations between them are seen as a conjunction (AND). As for a single institution, we have introduced three compliance relations between regulative norms in the specification of regulative norm nets, i.e., AND, OR and OE. Therefore, with an event occurring, when two (or more) regulative norm nets with contradictory compliance evaluations are combined, the conflicting status of the combined regulative norm net also depends on the compliance relations of the component regulative norm nets. If the compliance relation is AND, a conflict occurs since the combined regulative norm net cannot reach an agreement on the compliance of the event. If the compliance relation is OR, there is no such conflict since the combined regulative norm net only picks up the positive evaluation result, i.e., satisfied. When the compliance relation is OE, there is never a conflict since the activation period of the two component regulative norm nets will never overlap, since only when the origin is violated can the reparation be activated. Therefore, a norm conflict between two regulative norm nets may occur only when the two regualtive norm nets have an AND compliance relation.

Based on the description above, we define a *norm conflict* as follows.

Definition 6.6 (Norm conflict) *A norm conflict arises in a collective institution set* $\mathcal{C}_k = \{\widehat{NN}^i_1, \ldots, \widehat{NN}^i_m\}$ *with respect to the occurrence of an event* $\widehat{e}$ *iff either*

1. $\exists\ \widehat{NN}^i_x, \widehat{NN}^i_y \in \mathcal{C}_k$ *such that* NN^i_x *is evaluated to be satisfied and* NN^i_y *is evaluated to be violated, or*

2. $\exists\ \widehat{NN}^i_j \in C_k$ *and* $\exists\ RNN_x, RNN_y \in S_{RNN}(NN_j)$ *such that (1)* RNN_x *and* RNN_y *have an AND compliance relation, and (2)* $(RNN_x, CNN_{NN_j})^i$ *is evaluated to be satisfied and* $(RNN_y, CNN_{NN_j})^i$ *is evaluated to be violated.*

The first condition indicates the situation where a norm conflict involves two different institutions: with respect to the occurrence of the event $\widehat{e}$, the normative states of any two norm net instances from the collective institution set $\mathcal{C}_k$ are respectively evaluated to be satisfied and violated. The second condition indicates the situation where a norm conflict occurs within a single institution: with respect to the occurrence of the event $\widehat{e}$, there are two AND-related regulative norm nets in a norm net instance from the collective institution set $\mathcal{C}_k$, whose normative states are respectively evaluated to be satisfied and violated. In both conditions, an important requirement is that there are two component (regulative) norm nets that are respectively evaluated to be satisfied and violated with respect to the occurrence of an event. As for the second condition, another requirement is added, i.e, the compliance relations between the two components is AND.

It can be seen that our definition of a norm conflict is specified from the perspective of norm compliance, which is different from other definitions provided in the literature, e.g., [207] and [140]. This definition covers both weak and strong conflicts illustrated at the beginning of this section, since both types of conflicts have the feature of contradictory compliance evaluation results. Moreover, we take into account of the impact of the compliance relations between norms. Though the approach presented in [204] also specifies norm conflict through the analysis of norm compliance, the compliance relations between norms are not considered. Another thing to be noticed is the ontology of norm specification. For example, it is possible that an agent is regulated by both an obligation to sit and another obligation to stand whose activation period have an overlap. In this case, a conflict occurs since the two actions "sit" and "stand" are physically exclusive to each other. Therefore, if defining a norm conflict at the level of norm specification between an obligation and prohibition, such conflicts may not be covered.

Given the definition above, we now illustrate how to make use of the CPN models of NNs to computationally detect the norm conflicts. To do this, we first need to obtain the CPN mapping of a given norm net and the initial marking. Then we can calculate the state space of the CPN model according to the occurrence of events, by which we are able to examine the existence of specific patterns of CPN markings to indicate whether there are norm conflicts. Algorithm 3 gives a brief procedure of norm conflict detection in an instance of a norm net captured by a marked CPN model with respect to the occurrence of an event e. The complete algorithm is provided in Appendix A.8.

The algorithm mainly consists of three steps. The first step is to construct the CPN model of the norm net NN^i being checked and the result is a CPN mapping according to Definition 5.8. The second step is to obtain the new marking of the CPN model with respect to the occurrence of the enabled steps given the event e. Comparing the new marking with the previous one, we can derive the normative state changes of all the component regulative norm nets in NN^i by checking the changes of the marking of the places labeled with S (P_s) and the places labeled with V (P_v). The third step, including two sub steps, is to check the changes of markings obtained from the second step to determine whether there is any norm conflict in NN^i with respect to the occurrence of the event e. The first sub step checks whether NN^i is evaluated to be violated. If so, the second sub step further checks whether there are any two component regulative norm nets in NN^i that are respectively evaluated to be satisfied and violated, according to the changes in the marking of the satisfied and violated places.

Up to now, we have shown the mechanism of detecting norm conflicts by means of the CPN models of NNs. However, it is still not clear whether a norm conflict found in a norm net is a weak or strong conflict in the sense that a weak conflict can be avoided while a strong conflict cannot. To this end, we assume all the possible instances $\aleph$ of a norm net (i.e., all the possible states of the real system)

Algorithm 3 Conflict Query: CFQ

Require: (NN^i, e) ▷ A norm net instance and an external event
Ensure: CFS ▷ Conflicting status

▷ Obtain the new marking of a CPN model $\mathcal{N}$ given the occurrence of an enabled step Y with the current marking M
function UPDATESTATE($Y, M, \mathcal{N}$)
  ▷ The body of this function is presented in Appendix A.7
return M'
end function

  ▷ (**Step 1**) Obtain the CPN mapping of the norm net instance NN^i
$\langle(\mathcal{N}, P_s, P_v, P_{vr}, p_{so}, p_{vo}), M\rangle \leftarrow \Theta(NN^i)$

  ▷ (**Step 2**) Obtain the new normative state of NN^i after the event e occurs
$M' \leftarrow$ UpdateState(EStep($e, \mathcal{N}, M$)$, M, \mathcal{N}$)

  ▷ (**Step 3**) Check the normative state changes of all the component regulative norm nets in NN^i
    ▷ (**Step 3.1**) Check whether the normative state of NN^i is evaluated to be violated with respect to the occurrence of the event e
$CFS \leftarrow false$
if $M'(p_{vo}) - -M(p_{vo}) > 0$ **then**
  for all $(p_s, p_v) \in P_s \times P_v$ **do**
    ▷ (**Step 3.2**) Check whether there are two component regulative norm nets in NN^i such that one is evaluated to be satisfied and the other is evaluated to be violated with respect to the occurrence of the event e
    if $(M'(p_s) - -M(p_s)) > 0$ **and** $(M'(p_v) - -M(p_v)) > 0$ **then**
      $CFS \leftarrow true$
    end if
  end for
end if

and give the following definition.

Definition 6.7 (Weak conflict) *A weak conflict is detected in a norm net NN with respect to an event e iff* (1) $\exists NN^i \in \aleph : CFQ(NN^i, e) = true$, *and* (2) $\exists NN^{i'} \in \aleph : CFQ(NN^{i'}, e) = false$.

The first condition indicates that there exists a norm net instance of NN in which a norm conflict is found with respect to the occurrence of the event e. The second condition indicates that there exists a norm net instance of NN in which no norm conflict is found with respect to the occurrence of the event e.

Similarly, we give the definition of a strong conflict as follows.

Definition 6.8 (Strong conflict) *A strong conflict is detected in a norm net NN with respect to an event e iff* $\forall NN^i \in \aleph : CFQ(NN^i, e) = true$.

The condition of a strong conflict indicates that for every possible instance of the norm net NN there is always a norm conflict found with respect to the occurrence of the event e.

Given the definition of a weak conflict and strong conflict, we define a consistent norm net as follows.

Definition 6.9 (Consistent norm net) *A norm net NN is consistent iff $\forall e \in Ev_{NN}$, neither a weak conflict nor a strong conflict is detected in NN.*

A norm net is consistent if and only if the occurrence of any external event defined in the norm net does not lead to a norm conflict (weak and strong).

The problem of detecting a weak conflict in a norm net by means of the CPN model of the norm net is shown to be linear in the size of the possible norm net instances. In the worst case, however, we have to search over all the instances in the CPN model of the norm net. The problem of detecting a strong conflict is also shown to be linear in the size of the possible norm net instances, but always needs to go over all the instances in the CPN model of the norm net.

In this section, we have shown the definition of norm conflicts in the setting of collective institutions, and presented a mechanism of detecting norm conflicts by means of NNs and the CPN mapping. The norms specified by different institutions are assumed to be at the same level of abstraction. However, in practice, norms can be specified at different abstraction levels. For example, in a law system, there are typically layers of legislation and rule-giving from the constitutional level to the most concrete level of jurisprudence. To this end, in the next section, we extend the study of norms to multiple levels and discuss how norms specified at different levels are linked.

6.4 Norm Contextualization

In real world domains, norms are not specified at a single level of abstraction. For example, usually laws are first issued at a higher abstraction level stating the *do*s and *don't*s in directing the actors' behavior. Based on this abstract set of norms, elaboration will be conducted according to the specific characteristics and requirements of more concrete situations, which results into sets of contextual specifications of norms. This elaboration process facilitates a detailed explanation of the abstract norms in various application environments.

Following [117], we generalize three layers for modeling norms from abstract statements to concrete operations in terms of NNs. Firstly a general norm net is built to describe the expectations and boundaries for agent behavior in an abstract way. At this layer, norm specification is assumed to be stable through the life cycle of organizations. The second layer identifies the possible application contexts with their differing requirements by which a set of contextual norm nets are derived and present specific customizations to the general norm specification. At this layer, norm specification is bound with a certain context and is a reflection of the particular needs of that context. Moreover, a contextual norm net can again refers to sets of contextual norm nets in a recursive manner, which enables a flexible normative structure and facilitates norm designing at multiple abstraction levels.

Before presenting the formalization of norm contextualization, we first introduce the following definitions.

Definition 6.10 (Equal relation of regulative norm nets) *A regulative norm net RNN is equal to another regulative norm net RNN', denoted as $RNN = RNN'$, iff (1) $CR_{RNN} = CR_{RNN'}$, $LT_{RNN} = LT_{RNN'}$, and $RT_{RNN} = RT_{RNN'}$.*

Definition 6.11 (Subsumption relation of regulative norm nets) *A regulative norm net RNN is an subsumption of another regulative norm net RNN', denoted as $RNN \subseteq RNN'$, iff (1) $RNN = RNN'$, (2) $RNN \subseteq LT_{RNN'}$, or (3) $RNN \subseteq RT_{RNN'}$*

Norms in OperA+ are captured by the normative structure which is formalized by means of NNs. Therefore, we give the definition of contextualization of normative structure as follows.

Definition 6.12 (Contextualization of normative structure) *A contextualization of a normative structure is defined as a tuple (ctx, NN, NN') where*

- *ctx is a context as described in Definition 4.6,*
- *NN is a norm net, being referred to as general norm net,*
- *NN' is a norm net, being referred to as contextual norm net, such that*
  - $RNN_{NN} \subseteq RNN_{NN'}$,
  - $CNN_{NN} \subseteq CNN_{NN'}$,
  - $\forall \varepsilon \in Iv_{NN}, \exists n^c \in CNN_{NN'} : \nu_{n^c} = \varepsilon$,
  - $(AND(RNN_{NN}, RNN_{NN'}), CNN_{NN} \cup CNN_{NN'})$ *is a consistent norm net.*

The contextualization of a normative structure is defined between two norm nets such that the contextual norm net extends the general norm net by elaborating its consisting regulative norms and constitutive norms. The link between the concepts defined in these two norm nets is realized by the constitutive norms (counts-as relations) defined in the contextual norm net. For example, when an institutional event (*Staff_member*, *declare_cost*) defined in the general norm net is refined in the contextual norm net as (*PhD_student*, *declare_cost*), a constitutive norm needs to be defined in the contextual norm net to link the agents enacting the role of *PhD_student* to the role of *Staff_member* such that the norms imposed on *Staff_member* can be applied to the agents. In particular, the refinement relations between roles can be obtained from the contextualization of social structure presented in Section 4.2.4, which may serve as a source for norm contextualization. Moreover, the contextual norm net should be consistent with the general norm net, i.e., there is no norm conflicts in the two norm nets.

A reflection of norm contextualization can be found from laws and regulations in practice. The normative concepts in a refined context are more concrete. For example, whether a document should be considered as a required certificate in international trade depends on the context in which the concept of *certificate* is used. A required certificate for importing fruit from China to the EU might not *counts as* a required certificate for importing textile.

In general, a norm net can have multiple contextualizations with respect to different context refienment relations, while different norm nets may be referred to in one contextualization. Moreover, there is no clear boundary between two contexts, i.e., the contexts of different norm nets may overlap. For example, a context of the regulations for importing goods from Asia and another context of the regulations for importing textile products.

Finally, at the third layer, based on the contextual norm nets which contain enough information of the *do*s, *don't*s and *sub-ideals* in specific situations, the norms will be extended with operational aspects to capture the operational meaning of the norms such as how the violation is detected (detection mechanism), and what can be done by the organization to repair the violation and minimize the negative influence[7]. Actors only need to reason about the norms at the most concrete level but the process of contextualization helps them to identify the applicable norms according to their situated environment.

6.5 Conclusions

The design of normative structures of OperA+ models is extended in this chapter. We first introduce the concept of governance scope such that we can extend the institution model to capture the control boundary of an institution. The formalization of governance scope makes use of the context model presented in Chapter 3, which in turn reflects the contextual feature of OperA+. We further illustrate the coordination of multiple institutions in a virtual organization setting, which enables us to elaborate the analysis of how multiple institutions regulate organizational behavior.

It is possible that the norms in an organizational specification are not consistent, especially when they are from different institutions. This is a critical problem for organizational compliance since there might be no correct answer in determining whether or not an interaction structure is compliant if there are potential conflicts in the normative design itself. For this purpose, we provide a study of the problem of norm conflicts in terms of NNs. We differentiate between the concept of a potential conflict and a conflict, in which the former may be avoided and achieve a compliance while the latter cannot be avoided and will always lead to a violation. Importantly, we also consider the influence of compliance relations between norms in the formation of norm conflicts, which provides an integrated view on the analysis of a set of interrelated norms. To operationalize conflict detection, we make use of the CPN mapping of NNs and illustrate how to use state space of CPNs in detecting (potential) conflicts.

Norm contexutalization is another extension of normative design considered in this chapter, which expand the normative structure of an OperA+ model vertically, i.e., from abstract to concrete. We embed context refinement processes with elaboration of NNs such that abstract normative concepts can be linked to more concrete ones with respect to differing application environments (contexts). This not only provides a modular approach for building normative structures that improves both reusability and flexibility, but also enables actors to have a better understanding of their *do*s and *don't*s in changing environments.

Chapter 7

Preference Structure

This Chapter has the following contributions:

- a formalism for modeling agent preferences,
- a computational method for evaluating preference satisfaction, and
- a framework for combined reasoning with agent preferences in the context of norm compliance.

In the previous two chapters, we have shown the normative structure of organizational models, which is used to guide and regulate the behavior of participants towards the achievement of organizational objectives. Participants (human, software, or organizations) are the individual actors upon whom organizations depend to achieve their objectives. They are expected to follow the normative constraints that are imposed on them through the roles they enact in the organization. However, participants are autonomous entities who are capable of controlling their own actions. The autonomy nature of the participants implies that they have the tendency to act upon individual interests and needs, which is important to ensure personal satisfaction and involvement. To this end, it is necessary to take into account the autonomy reflected in the preferences of the participating actors together with the conformity brought in by the normative structure in determining the interaction structure of an organization model.

From a global perspective, normative constraints are imposed to direct the behavior of the actors to ensure the achievement of organizational objectives. From an individual perspective, the autonomy of individual actors indicates their personal preferences, and influence their choices of actions to fulfill their roles

This chapter is based on our contribution to the 13th International Conference on Autonomous Agents and MultiAgent Systems, pp. 1373–1374, 2014 [122].

in the organization. These preferences could be conceived of as an individual's attitude towards a set of alternatives of actions. Considering both perspectives, we argue that organizational compliance should go beyond norm compliance to consider the satisfaction of the actors' personal preferences. This requires the representation of both normative constraints and personal preferences to evaluate the effectiveness of possible interaction structures.

This chapter is organized as follows. In the next section, we discuss how the notion of preference is used in agent systems. Section 7.2 introduces our framework for combined reasoning with norms and preferences. In the following three sections 7.3, 7.4 and 7.5, we illustrate in detail the three modules consisting in the framework. Finally, we conclude this chapter in Section 7.6.

7.1 Preferences in Agent Systems

In the research of agent systems, the notion of preference has been introduced to facilitate the decision making of agents when alternative plans are available. For example, in [210], preferences have been incorporated into the BDI execution process of agents and used to guide the choices made. In [105], temporal logic is used for the representation of various kinds of goals and preferences, in which the former are considered as hard constraints and the latter soft constraints for choosing a most preferred action among the available ones. To address the problem of conflicting guidance for agents, Myers and Morley combined a guidance-based preference relation over plan choices with an ability to extend the set of options considered by an agent when conflicts arise [160] .

A number of preference models for agents, e.g., [210] and [105], has been inspired by the research in the planning domain where users' preferences are considered to generate high quality plans. For example, in [23]), Bienvenu et al. proposed a logical language for specifying preferences over the evolution of states and actions associated with a plan and provided a semantics for the preference language in the situation calculus. In [197], Son and Pontelli presented a declarative language for the specification of preferences between possible solutions of a planning problem, together with an implementation using answer-set programming. Delgrande et al. developed a framework for characterizing preferences and properties of preference-based planning [50]. The preference language they propose is propositional and distinguishes between choice preferences and temporal preferences.

In this research, we extend the existing work of modeling agent preferences [210] and [23], for the purpose of representing preferences of individual participants in the setting of organizational interactions. More importantly, we investigated the integration of preference reasoning with the evaluation of norm compliance such that we are able to find a balance between organizational control and individual autonomy. In the following sections, we give a detailed illustration of our approach of modeling individual preferences and evaluating preference satisfaction in the context of norm compliance.

7.2 Reasoning with Preferences in Normative Systems

Organizational compliance should go beyond norm compliance to consider the satisfaction of the actors' personal preferences, which gives rise to the need of representing both normative constraints and personal preferences to evaluate the effectiveness of organizational interactions in a unified way.

To this end, we propose a framework for combined reasoning with agent preferences in the context of norm compliance, as shown in Figure 7.1. It consists of three modules: specification, evaluation and combined-reasoning. The *specification module* presents the normative constraints relevant to the control of the organization, the preferences of the participating agents in the organization, and the possible interaction plans of the agents. Given the specification module, the *evaluation module* will evaluate each interaction plan following two parallel steps: (1) the *compliance evaluation* is to verify the plan against the normative constraints and determine the compliance status of the plan, and (2) the *satisfaction evaluation* is to verify the plan against the preferences of each agent and indicate to what extent the agent is satisfied with the plan. As a result, we obtain an integrated picture of the compliance status and the satisfaction level of all the participating agents with respect to each plan. Finally, in the *combined-reasoning module*, each agent uses its own *combined-reasoning* strategy to determine the combined effects of each plan with the information obtained from the evaluation module. As such, we are able to derive the effectiveness of each plan in achieving a compliance balanced between norm compliance and preference satisfaction.

In the following sections, we illustrate each of these three modules in more detail, with a travel case as an illustrating scenario.

7.3 Specification Module

In the specification module, we need to formalize the normative constraints, agent preferences and interaction plans. For normative constraints, we use the normative language NNs presented in Chapter 5. To formalize agent preferences, we introduce a preference language based on the approaches presented in [210] and [23]. The interaction plans are captured using the formalism (Definition 5.10) presented in Chapter 5.

7.3.1 Preference Language

In this subsection, we present an extension of the preference language proposed in [210] and [23] with the following three aspects:

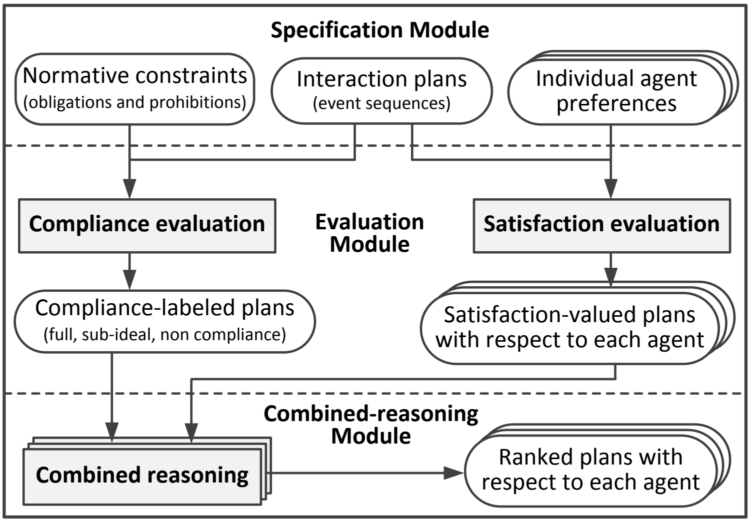


Figure 7.1: *Combined reasoning framework*

(1) an agent's preferences are specified over actions. For example, *Carl* prefers the action of *taking a plane* over the action of *taking a train*.

(2) agents' preferences are not restricted to its own actions but might depend on other agents' actions. For example, *Carl* has a preference of *taking a plane* over *taking a train* if the other two members *Alice* and *Bob* in the group choose to *take a plane* (**example1**).

(3) an agent's preference might depend on the actions of some other agents enacting a particular role in the organization. For example, *Carl* has a preference of *taking a plane* over *taking a train* if the *leader* of the group chooses to *take a plane* (**example2**).

It can be seen that agent preferences are specified over sequences of actions relating to specific agents or roles. In this sense, preferences are not isolated but have an interactive nature, which enables a richer expressiveness of how the agents perceive themselves in an organizational setting where collective objectives need to be achieved through their interactions.

Following the naming conventions in [210] and [23], we use *basic desire formulas* to represent basic statements about the preferred actions, *atomic preference formulas* to represent an ordering over basic desire formulas and *general preference formulas* to express atomic preference formulas that are optionally subjected to a condition. The preferences of an agent are specified as a set of general preference formulas.

Definition 7.1 (Basic desire formula) *A basic desire formula is defined as a tuple (ag, pr, $flag$) where*

- *ag is an agent identifier, indicating to whom this preference belongs,*
- *$pr \in L_E$ is an event formula (a combination of actions relating to some agents or roles), indicating the content of this preference, and*
- *$flag \in \{true, false\}$, indicating whether pr is preferred ($true$) or not preferred ($false$).*

By using event formulas, we can express preferences of an agent over sets of actions with various relations (absence, conjunction, disjunction, sequence). For example, one may have a preference of checking in at the hotel before attending the meeting which indicates a preference over a sequence of actions. Specifically, $flag$ is used to specify whether some actions are preferred or not preferred. For example, one may have a preference of not taking a plane, with the attribute of *flag* setting to be "false". In this sense, if there is no action of taking a plane occurred, the preference is satisfied. Note that we allow the preference of an agent to be specified over the actions of other agents as well as roles, i.e., the agent could prefer some other agents (enacting specific roles) performing some actions. For example, the group leader might have a preference that the group members take a train for the travel. In this case, if all the group members take the action of taking a train, the preference of the group leader is satisfied.

Basic desire formulas are articulated over a single affair and provide the building blocks of our preference language in terms of event formulas. While, in practice, preference are more often stated over several affairs to indicate the differing degrees of favor among several alternatives. To represent such preferences, we give the definition of an atomic preference formula.

Definition 7.2 (Atomic preference formula) *Let v_{min} and v_{max} be numeric values such that $v_{min} < v_{max}$. An atomic preference formula is of the form*
$\eta_0(v_0) \gg \ldots \gg \eta_n(v_n)$, $n \geq 0$, where

- *$\eta_i = (ag, pr, flag)$ is a basic desire formula,*
- *$\forall \eta_i, \eta_j, 0 \leq i, j \leq n, i \neq j : (ag)\eta_i = (ag)\eta_j$, and*
- *$v_{min} \leq v_i \leq v_{max}$ and $v_i < v_j$ for $i < j$.*

An atomic preference formula presents a preference of the same agent (indicated by the second condition) over a set of alternatives η_i, each of which is assigned a value indicating the cost of taking a specific alternative. Thus the lower the index i is the more preferred the alternative is, i.e., the less the cost is the more the alternative is favored. In addition, it is the least preferred when none of these alternatives are satisfied. Following [210], we use $v_{min} = 0$ and $v_{max} = 100$. An example of an atomic preference formula is (*Alice*, (*Alice*, *take_train*), *true*)(20)

$\gg$ (*Alice*, (*Alice*, *take_plane*), *true*)(60). This indicates that *Alice* prefers taking a train over taking a plane, and the degrees of preference are 20 and 60 respectively.

Another type of preferences is conditional preferences, i.e., only when some conditions hold the preference is activated. For example, one may prefer taking a plane over taking a train if the other partner takes a plane. In order to specify such preferences, we give the definition of a general preference formula to express atomic preference formulas that are optionally preceded by a condition.

Definition 7.3 (General preference formula) *A formula ϕ is a general preference formula if one of the following holds:*

- *ϕ is an atomic preference formula,*
- *ϕ is $\gamma : \Psi$, where γ is an event formula (called condition), and Ψ is an atomic preference formula,*

Notice that, in general preference formulas, conditions are built over event formulas. That is, agent preferences may rely on either the actions of specific agents or the actions relating to particular roles. For example, we can express the examples **example1** and **example2** given at the beginning of this subsection as the following general preference formulas.

- conditioned on the actions of some agents, the first formula is (*Alice*, *take_plane*) $\wedge$ (*Bob*, *take_plane*): (*Carl*, (*Carl*, *take_plane*), *true*)(v_1) $\gg$ (*Carl*, (*Carl*, *take_train*), *true*)(v_2).
- conditioned on the actions relating to a role, the second formula is (*leader*, *take_plane*): (*Carl*, (*Carl*, *take_plane*), *true*)(v_3) $\gg$ (*Carl*, (*Carl*, *take_train*), *true*)(v_4).

In the preference specification, an issue is worth discussing. Assume that *Bob* has a preference described as (*Bob*, (*Bob*, *take_train*), *false*)(20) $\gg$ (*Bob*, (*Bob*, *take_plane*), *true*)(30), indicating that Bob has a preference of not taking a train over taking a plane, and suppose there are two candidate plans (*Bob*, *take_plane*) and (*Bob*, *drive*). Since both plans satisfy the preference (*Bob*, (*Bob*, *take_train*), *false*), they both obtain a satisfaction of 20. However, the first plan also satisfies the preference (*Bob*, (*Bob*, *take_plane*)), which obtains a satisfaction of 30. According to [210] and [23], we should select the minimum value 20 for the first plan and thus the two plans are at the same satisfaction level. This contradicts the intuition that the first plan should be more preferred since *Bob* has an explicit preference of taking a plane. In such cases, we might need more elaborated metrics for the satisfaction evaluation.

Given the preference formulas defined above, we capture the preferences of orgarnizational participants and thereby construct the preference structures of OperA+ models. Formally, we give the definition of a prefernce structure as follows.

Definition 7.4 (Preference structure) *A preference structure PS is defined as a set of general preference formulas* $\{\phi_1, \phi_2, \ldots, \phi_n\}$.

A preference structure consists of a set of preference formulas which are related to different agents participating in the organization. It is possible that some of the agents do not disclose their preferences and other agents have more than one preferences. Thus, the set of preference formulas may not cover the preferences of every agent in an organization.

The link between the preference structure and the social structure of an organizational model is the set of roles and agents specified in the social structure and the set of preferences defined over the actions of the agents and roles in the preference structure.

7.3.2 Contextualization in Agent Preferences

As described in Chapter 3, organizations place varying compliance requirements on agent behavior with respect to different contexts, which can be reflected from the construction of social relations and normative constraints in OperA+. In this sense, agents need to understand the context where they are situated and adapt their behavior and decision making strategies accordingly.

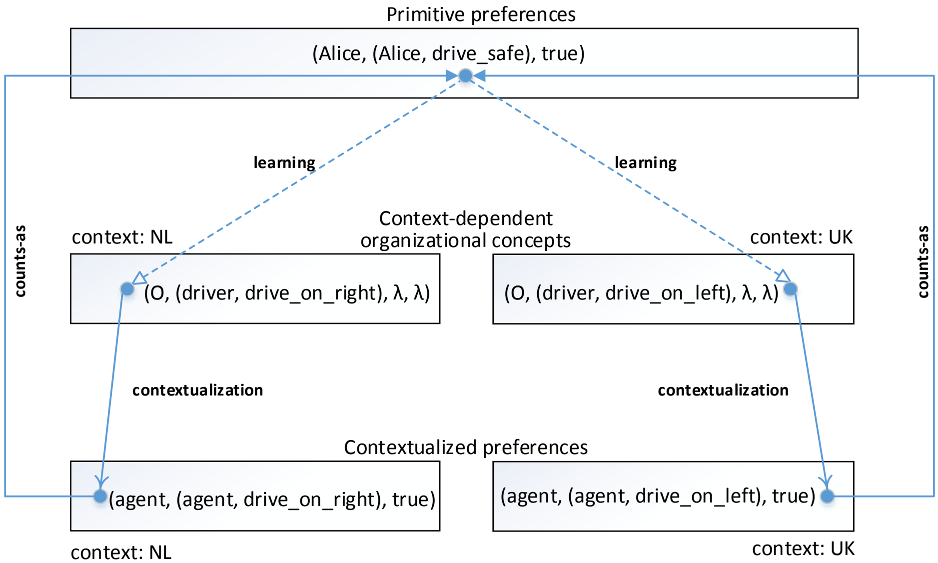


Figure 7.2: *Contextualization of agent preferences.*

Agents themselves, not necessarily participating in a specific organization, may have some primitive preferences, indicating their abstract values and interests.

When the agents come into a concrete context, the abstract preferences of the agents need to be linked to the concrete concepts used in the context. For example, an agent *Alice* has a primitive preference of driving safely. Assume we have two contexts $ctx_1 = NL$ and $ctx_2 = UK$ in terms of the geographical difference. In ctx_1 there is a regulation saying that drivers should drive along the right side of the road ($drive\_on\_right$), while in ctx_2 it is prescribed that drivers should drive along the left side of the road ($drive\_on\_left$). When the agent comes into ctx_1 or ctx_2, the agent has to associate its abstract preference of driving safely to the obligation of driving on the right or driving on the left, generating two different contextual preferences. Therefore, with respect to different contexts, agents have to contextualize their primitive preferences such that they can reason about the organizational concepts in differing contexts.

The process of contextualizing abstract preferences can be seen as preceded by a learning process of agents when they come into different contextual settings. This largely exist in our everyday life, e.g., when we move into a new country, study in a new university, join in a club. We have to adapt ourselves to the new environment, and apply the contextual information in the new environment to refine our abstract preferences. In order to formalize the contextualization of agent preferences, we adopt the counts-as rules proposed by Searle [188], which have been widely studied by researchers to connect abstract concepts to more concrete ones. Figure 7.2 shows the outline of the preference contextualization framework through an example.

7.4 Evaluation Module

The evaluation module (cf. Figure 7.1) consists of two operations: compliance evaluation and satisfaction evaluation. For the compliance evaluation, we use the CPN-based computational model presented in Chapter 5. For the satisfaction evaluation, we extend this computational model and propose a set of mapping rules to operationalize the aggregation evaluation of individual preferences.

7.4.1 Operationalization of Preference formulas

Preference formulas are built over event formulas using numeric values to indicate different levels of preferences. In general, there are two types of information that needs to be represented: (1) the events constituting a preference formula and the relations between the events as shown in Definition 5.2, and (2) different levels of satisfaction and more importantly the evaluation of agent actions with respect to different types of preference formulas, i.e., under what conditions a preference is satisfied and to what extent the preference is satisfied (indicating by the satisfaction value).

In the following, we illustrate by examples how such information is captured in Colored Petri Nets with respect to different types of preference formulas. In graphical representation, agents are indicated by the black dots with the information carried by the agents displayed next to the dots, the types of the agents are indicated by the kinds of information the agents can carry (e.g., name, role, and satisfaction value), actions are indicated by rectangles, the availability of the actions to the agents are indicated by the arcs going from the circles to the rectangles, and the changes of the agents' satisfaction values are indicated by the expressions above or below the arcs.

The color sets defined for the CPN mapping of preferences are listed in Table 7.1. The color set *Agent* is defined as *string* and the values are the names of agents. Each role r is defined as a different color set through the operator *with* and the name of the role. The color set *satis* is *int* and the values are integers, indicating different levels of preference satisfaction. Agents enacting a specific role are represented by the color set *r_ag* which is a product of the color sets *Agent* and r. Similarly, the satisfaction of agent is represented by the color set *agent_satis* which is a product of the color sets *Agent* and *satis*, indicating the an agent's satisfaction level.

Color Sets	**Example**	**Explanation**
colset Agent = string;	"Alice"	an agent named "Alice"
colset r = with $name_r$;	Group_leader	a role named Group_leader
colset satis = int;	20	a satisfaction level *20*
colset r_ag = product r ∗ Agent;	(Group_leader, "Alice")	the agent "Alice" enacting the role Group_leader
colset agent_satis = product agent ∗ satis;	("Alice", 20)	the agent "Alice" has a satisfaction level of *20*

Table 7.1: *Color sets defined for the mapping of agent preferences*

Basic desire formulas: Figure 7.3(a) gives the graphical representation of the preference formula η_0, expressing that *Bob prefers taking a plane.* Initially, the agent *Bob* situated in the left circle, has not performed the action *take_plane.* Accordingly his preference η_0 is not satisfied, which is indicated by the satisfaction value *100* (v_{max}) carried by the token in the right circle (labeled *SE*). After performing the action, *Bob* moves to the place *SE* and his satisfaction value changes to *0* (v_{min}), indicating that his preference η_0 is satisfied. *SE* represents the Satisfaction Evaluation of the corresponding preference formula by extracting different satisfaction values attached to an agent, i.e., *0* (satisfied) or *100* (not satisfied). Note that for simplicity, we assume here a complete knowledge of the domain options (e.g. traveling by car is not traveling by plane). In Figure 7.3(b), η_1 represents that *Bob prefers* ***not*** *taking a train.* Initially, the satisfaction value of *Bob* is set to *0*, indicating that η_1 is satisfied. When the event (*Bob*, *take_train*) occurs, the satisfaction value changes to *100*, indicating that η_1 is not satisfied any more. Algorithm 4 shows the procedure of generating the CPN model of a

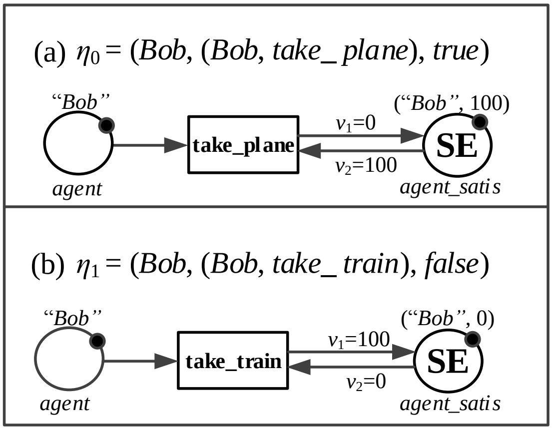


Figure 7.3: *CPN pattern of a basic desire formula*

given basic desire formula.

Algorithm 4 CPN mapping of a basic desire formula: BDF_CPN

Require: $(ag, pr, flag, v_{min}, v_{max})$
Ensure: $(\mathcal{N}, p_{\text{SE}})$
▷ obtain the CPN model of the event formula pr
$\mathcal{N} \leftarrow$ EventFormula_CPN(pr)

▷ update the color of the sink place of the CPN model $\mathcal{N}$
$colset\ satis = int$
$colset\ agent\_satis = product\ Agent * satis$
$C_{\mathcal{N}} \leftarrow C_{\mathcal{N}} \cup (SKP_{\mathcal{N}}, agent\_satis)$

▷ add the satisfaction info depending on the flag component
$A_{\mathcal{N}} \leftarrow A_{\mathcal{N}} \cup (SKP_{\mathcal{N}}, IT_{SKP_{\mathcal{N}}})$
if $flag = true$ **then** ▷ preferred
  $M(SKP_{\mathcal{N}}) \leftarrow (ag, v_{max})$
  $E(IT_{SKP_{\mathcal{N}}}, SKP_{\mathcal{N}}) \leftarrow (ag, v_{min})$
  $E(SKP_{\mathcal{N}}, IT_{SKP_{\mathcal{N}}}) \leftarrow (ag, v_{max})$
else ▷ not preferred
  $M(SKP_{\mathcal{N}}) \leftarrow (ag, v_{min})$
  $E(IT_{SKP_{\mathcal{N}}}, SKP_{\mathcal{N}}) \leftarrow (ag, v_{max})$
  $E(SKP_{\mathcal{N}}, IT_{SKP_{\mathcal{N}}}) \leftarrow (ag, v_{min})$
end if
$p_{\text{SE}} \leftarrow SKP_{\mathcal{N}}$

Atomic preference formulas: In Figure 7.4, the model of an atomic preference formula is presented, which consists of the two basic desire formulas η_0 and

η_1. With values of *20* and *30* respectively defined on η_0 and η_1, their levels of

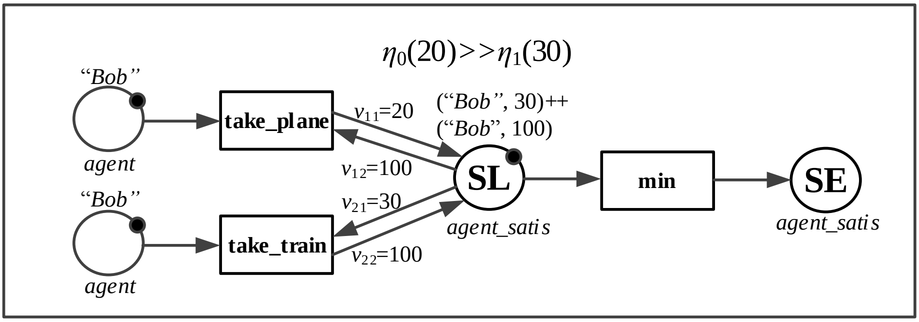


Figure 7.4: *CPN pattern of an atomic preference formula*

preferences are differentiated. The combination of these two basic desire formulas is featured by a place labeled *SL* which keeps a Satisfaction List of all the constituting basic desire formulas, i.e., which constituting preferences are satisfied and which are not. The satisfaction evaluation of the atomic preference formula will be the lowest satisfaction value in the place *SL*. To operationalize such a process, a construction *min* is used to transfer the lowest satisfaction value from *SL* to the Satisfaction Evaluation place *SE*. Algorithm 5 gives the procedure of generating the CPN model of a given atomic preference formula.

General preference formulas with conditions: For preference formulas, Figure 7.5 shows an example based on Figure 7.4. There are mainly three adjustments: (1) the condition φ is connected to both η_0 and η_1, indicating that only when the condition holds the two preferences can be Activated (corresponding to the place labeled *A*), (2) there is a satisfaction value *0* attached to the agent *Bob* in the place labeled *SL*, indicating that the preference indicated by $\varphi : \eta_0(20) \gg \eta_1(30)$ is not activated initially [1], and (3) the place *SL* is connected to the place *A* through a *null* transition such that when the condition holds (an agent situates in the place *A*) the agent with the satisfaction value *0* will be removed from *SL*. The connections from the place *A* are bidirectional arcs since the condition is needed by several actions to activate the satisfaction evaluation. The unlabeled action is an action that occurs as soon as it is enabled. Algorithm 6 presents the procedure of generating the CPN model of a given general preference formula.

[1] If a preference is conditional, it does not contribute to the aggregation evaluation when the condition does not hold and thus we add a token with the satisfaction value *0* initially.

Algorithm 5 CPN mapping of an atomic preference formula: APF_CPN

Require: $\{(\eta_0, v_0), (\eta_1, v_1), \ldots (\eta_n, v_n)\}$
Ensure: $(\mathcal{N}, p_{\mathrm{SL}}, p_{\mathrm{SE}})$
  ▷ build a CPN model for each basic desire formula contained in the atomic preference formula
  for $i = 0 \to n$ **do**
    $(ag, pr, flag, v_{min}, v_{max}) \leftarrow \eta_i$
    $(\mathcal{N}_i, p_{\mathrm{SE}_i}) \leftarrow \mathrm{BDF\_CPN}(ag, pr, flag, v_i, v_{max})$
  end for
  ▷ combine the CPN models and adjust the satisfaction information
  $C_{\mathcal{N}} \leftarrow C_{\mathcal{N}} \cup \{(p_{\mathrm{SL}}, C(p_{\mathrm{SE}_0}))\}$
  for $i = 1 \to n$ **do**
    $M(p_{\mathrm{SL}}) \leftarrow M(p_{\mathrm{SL}}) + +M(p_{\mathrm{SE}_i})$
  end for
  for $i = 0 \to n$ **do**
    $SKP_{\mathcal{N}_i} \leftarrow p_{\mathrm{SL}}$
  end for
  ▷ add a construction *min* such that the minimum value of the satisfied alternatives can be obtained
  $P_{\mathcal{N}} \leftarrow P_{\mathcal{N}_1} \cup \ldots \cup P_{\mathcal{N}_n} \cup \{p_{\mathrm{SE}}\}$
  $T_{\mathcal{N}} \leftarrow T_{\mathcal{N}_1} \cup \ldots \cup T_{\mathcal{N}_n} \cup min$
  $A_{\mathcal{N}} \leftarrow A_{\mathcal{N}_1} \cup \ldots \cup A_{\mathcal{N}_n} \cup \{(p_{\mathrm{SL}}, min), (min, p_{\mathrm{SE}})\}$
  $\Sigma_{\mathcal{N}} \leftarrow \Sigma_{\mathcal{N}_1} \cup \ldots \cup \Sigma_{\mathcal{N}_n}$
  $C_{\mathcal{N}} \leftarrow C_{\mathcal{N}_1} \cup \ldots \cup C_{\mathcal{N}_n} \cup \{(p_{\mathrm{SE}}, C(p_{\mathrm{SL}}))\}$

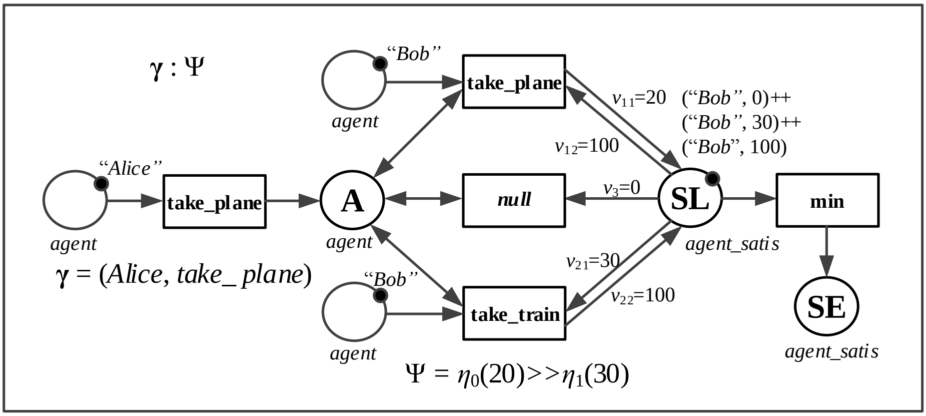


Figure 7.5: *CPN pattern of a general preference formula*

Algorithm 6 CPN mapping of a general preference formula: GPF_CPN

Require: (γ, Ψ)
Ensure: $(\mathcal{N}, p_{\text{SE}})$
  ▷ create the CPN model of the condition γ
  $\mathcal{N}_\gamma \leftarrow$ EventFormula_CPN(γ)

  ▷ create the CPN model of the atomic preference Ψ
  $(\mathcal{N}_\Psi, p_{\text{SL}}, p_{\text{SE}}) \leftarrow$ APF_CPN(Ψ)

  ▷ combine the CPN models and adjust the satisfaction information
  $\{(\eta_0, v_0), \ldots (\eta_n, v_n)\} \leftarrow \Psi$
  $(ag, pr, flag, v_{min}, v_{max}) \leftarrow \eta_0$ ▷ obtain the name of the agent
  for all $p \in SCP_{\mathcal{N}_\Psi}$ **do**
    $A' \leftarrow A' \cup \{(SKP_{\mathcal{N}_\gamma}, OT_p), (OT_p, SKP_{\mathcal{N}_\gamma})\}$
  end for
  $M(p_{\text{SL}}) \leftarrow M(p_{\text{SL}}) + +(ag, v_{min})$
  $T_{\mathcal{N}_\Psi} \leftarrow T_{\mathcal{N}_\gamma} \cup T_{\mathcal{N}_\Psi} \cup null$
  $P_{\mathcal{N}_\Psi} \leftarrow P_{\mathcal{N}_\gamma} \cup P_{\mathcal{N}_\Psi}$
  $A_{\mathcal{N}_\Psi} \leftarrow A_{\mathcal{N}_\gamma} \cup A_{\mathcal{N}_\Psi} \cup A' \cup \{(p_{\text{SL}}, null), (SKP_{\mathcal{N}_\gamma}, null), (null, SKP_{\mathcal{N}_\gamma})\}$
  $\Sigma_{\mathcal{N}} \leftarrow \Sigma_{\mathcal{N}_\gamma} \cup \Sigma_{\mathcal{N}_\Psi}$
  $C_{\mathcal{N}} \leftarrow C_{\mathcal{N}_\gamma} \cup C_{\mathcal{N}_\Psi}$
  $E_{\mathcal{N}} \leftarrow E_{\mathcal{N}_\gamma} \cup E_{\mathcal{N}_\Psi} \cup \{((p_{\text{SL}}, null), (ag, 0))\}$

7.4.2 Aggregating Preference Formulas

Each agent can possibly have more than one preference, which leads to the question of how to aggregate the evaluation results of individual preferences relating to the same agent. In [23], the authors reviewed several aggregation methods. Among them, conjunction and disjunction are two simple aggregation methods, in which conjunction is to maximize the satisfaction of all the component preferences, whereas disjunction indicates that an agent would be content if any of the component preferences are satisfied. In cases where an agent's preferences are of greatly different levels of importance, an appropriate aggregation method is to use the standard lexicographic ordering, which can also be combined with conjunction and disjunction. In cases where the component preferences are considered equally important, a well-suited method is to firstly sort the levels of satisfaction of the component preferences in non-decreasing order and then apply the lexicographic ordering. In cases where the preferences are attached with numeric values to indicate different levels of importance, a simple method is to sum the levels of satisfaction of the component preferences, which amounts to maximizing the average level of satisfaction.

The preference language introduced in Section 7.3.1 uses numeric values to indicate different levels of importance (a value indicates the cost relating to an alternative). Following [210] and [23], an appropriate aggregation method is to sum the evaluation results of an agent's component preferences. Accordingly,

Figure 7.6 gives a CPN construction of this summation method, which is realized by taking the evaluation results of all the component preference formulas as an input. The aggregation result will be produced in the place labeled *PS* indicating the overall Preference Satisfaction of an agent. Algorithm 7 gives the procedure of generating the CPN model of aggregated preferences of an agent.

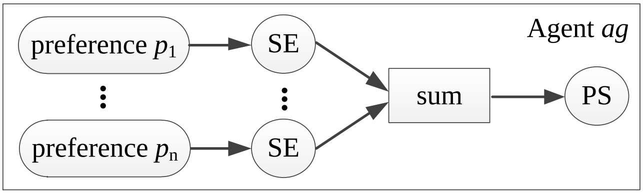


Figure 7.6: *CPN pattern of preference aggregation*

Algorithm 7 CPN mapping of aggregated preference formulas: AGPF_CPN

Require: $\{(\gamma_1, \Psi_1), \ldots, (\gamma_n, \Psi_n)\}$
Ensure: $(\mathcal{N}, p_{\mathrm{PS}})$
▷ Obtain the CPN mapping of each general preference formula
for all $i = 1 \rightarrow n$ **do**
  $(\mathcal{N}_i, p_{\mathrm{SE}_i}) \leftarrow$ GPF_CPN(γ_i, Ψ_i)
end for
▷ Aggregate all the general preference formulas
$T_{\mathcal{N}} \leftarrow T_{\mathcal{N}_1} \cup \ldots \cup T_{\mathcal{N}_n} \cup \{sum\}$
$P_{\mathcal{N}} \leftarrow P_{\mathcal{N}_1} \cup \ldots \cup P_{\mathcal{N}_n} \cup \{p_{\mathrm{PS}}\}$
for all $i = 1 \rightarrow n$ **do**
  $A' \leftarrow A' \cup \{(p_{\mathrm{SE}_i}, sum)\}$
end for
$A_{\mathcal{N}} \leftarrow A_{\mathcal{N}_1} \cup \ldots \cup A_{\mathcal{N}_n} \cup A' \cup \{(sum, p_{\mathrm{PS}})\}$
$\Sigma_{\mathcal{N}} \leftarrow \Sigma_{\mathcal{N}_1} \cup \ldots \cup \Sigma_{\mathcal{N}_n}$
$C_{\mathcal{N}} \leftarrow C_{\mathcal{N}_1} \cup \ldots \cup C_{\mathcal{N}_n} \cup \{(p_{\mathrm{PS}}, p_{\mathrm{SE}_0})\}$
$E_{\mathcal{N}} \leftarrow E_{\mathcal{N}_1} \cup \ldots \cup E_{\mathcal{N}_n}$

7.4.3 Evaluating Agent Interactions

Up to now, we have introduced all the formalisms for the construction of the social structure, normative structure, and preference strucuture of OperA+. Given an organizational model, the social structure specifies a set of interdependent roles, the normative structure specifies a set of norms, and the preference structure specifies a set of agent preference, which are used to regulate and direct the interactions of the participating agents. The interactions between the agents are captured by a set of interaction plans (sequences of agent actions). In order to evaluate agent interactions in an organization, on the one hand, we check whether the actions of the agents are complying with the norms according to the roles they enact, and on the other hand, we evaluate to what extent the preferences of the agents are satisfied.

In Chapter 5, we have shown the mechanism of checking norm compliance of agent interactions, i.e., whether the actions of the agents violate the norms imposed by the organization where the agents enact roles. In this section, we show how to include the evaluation of agent preferences in this mechanism such that we are able to reason about both aspects on agent interactions in a unified way. Firstly, we assume a set of norms captured by a norm net NN and a set of agent preferences captured by a set of preference formulas Φ. Accordingly, we construct a hierarchical CPN model of both NN and Φ. Given an interaction plan IP, compliance checking is achieved using the compliance query algorithm CLQ (Algorithm 2) presented in Chapter 5. The results are indicated by labeling each plan with compliant (C), sub-ideal compliant (C*) or violated (V), in which C* means that there are repaired violations. To realize the satisfaction evaluation, we extend the normative evaluation approach by integrating the mapping rules for agent preferences described in Section 7.4.1. Algorithm 8 gives a brief description of how the integrated model is used. The detailed procedure is shown in Appendix A.9.

Algorithm 8 Evaluation of norm compliance and preference satisfaction

Require: $(\langle(\mathcal{N}_H, P_s, P_v, P_{vr}, p_{so}, p_{vo}, P_{PS}), M\rangle, IP)$
Ensure: (CE, PS)
function UPDATEMARKING($Y, M, \mathcal{N}$)
  ▷ the body of this function is presented in Appendix A.9
end function
  ▷ Norm compliance and preference satisfaction
for $j = 1$ to $|IP|$ **do**
  $M' \leftarrow$ UpdateMarking(ESteps($IP[j], \mathcal{N}, M$), $M, \mathcal{N}_H$)
  ▷ (1) Compliance checking
  $CE[j] \leftarrow$ ComplianceEva(M, M', P_{vr}, p_{vo})
end for
  ▷ (2) Satisfaction evaluation
for all $p \in P_{\text{PS}}$ **do**
  $i \leftarrow 1$
  $PS[i] \leftarrow M'(p)$
  $i \leftarrow i + 1$
end for

The algorithm needs three inputs: a hierarchical CPN model $\mathcal{N}_H$ capturing a set of norms and a set of agent preferences, a marking M, and an interaction plan IP. The body of the algorithm consists of two major steps. The first step calls the function *ComplianceEva* to realize compliance checking. The second step realizes the satisfaction evaluation by checking all the places labeled *PS* to obtain the satisfaction value of all the agents, according to the updated state of the CPN model after the occurrence of each each event in the interaction plan.

7.5 Combined-reasoning Module

Up to now, we have shown the formalisms to capture normative constraints and agent preferences, and the mechanisms to evaluate norm compliance and preference satisfaction. However, as illustrated at the beginning of this chapter, these two sources of information need to be combined in the evaluation of agent interactions such that we can find a balance between organizational control and agent autonomy in the achievement of organizational compliance. There are two possible ways of combining such information. One is to specify preferences over compliance states, and the other is to specify different strategies of combining compliance and preference evaluation results. Given that we don't yet consider preferences over compliance states, here we introduce some examples of combination strategies at two different levels of reasoning. The first level is the agent level which shows the strategies that might be used by individual agents in terms of the possible types of the agents. The second level is the organization level which shows the strategies that might be adopted by organizations in terms of the types of social environments where the organizations live.

7.5.1 Agent Level

From an agent's perspective, whether a interaction plan is adopted depends on the combined-reasoning strategy of the agent by which the agent combines the two sources of information (norm compliance and preference satisfaction) to reason about the feasibility of the plan. We exemplify three categories of agents with their respective combined-reasoning strategies as follows:

(1) *Selfish* agents only care about their personal preferences and thus their strategy is to maximize their own satisfaction level without considering the normative consequence.

(2) *Norm-aware* agents are more concerned with norm compliance and thus their strategy is to maximize their own satisfaction level provided that all the norm are complied with.

(3) *Social* agents are more concerned with the social welfare and thus their strategy is to maximize the satisfaction level of the whole population.

With respect to their strategies, the individual agents can make the decisions on which interaction plan to execute. Notice that the agents will use such a reasoning process continuously according to the changes in the environment.

7.5.2 Organization Level

From an organizational perspective, norm compliance is an important feature for interaction plans since violations may cause a failure to the organization as a whole. Therefore, at the organization level, a preliminary strategy for combined reasoning is to minimize the violations in the organization. Furthermore, we can have different strategies in terms of the possible types of social environment where the organization is situated. Here we show two possible examples:

(1) In a *hierarchical* organization, the satisfaction of the agents with higher authority (indicating by the role relation in OperA) in the organization might be more important and therefore the reasoning strategy of the organization would be to maximize the preference satisfaction of such agents (usually associated with particular roles).

(2) In a *horizontal* organization (sometimes it is referred to as a flat organization), the agents are equal in status and thus the reasoning strategy of the organization would be to maximize the average satisfaction level of all the agents.

The organization-level combined-reasoning is based on the availability of the individual agents' preferences. It can serve as a macro-control on top of the individual agents' reasoning processes, which seeks a trade off among the differences of the autonomous agents' preferences from a system perspective.

7.6 Conclusions

In this chapter, we introduced a preference language for the construction of the preference structure of OperA+, which enables to capture the individual preferences of participants in organizations. Furthermore, we explored the integration of agent preferences with normative constraints in the evaluation of agent interactions, which aims to find a balance between organizational control and individual autonomy in achieving organizational compliance.

From an individual perspective, the combined reasoning approach can provide the necessary information for agents to make decisions considering both normative constraints and personal preferences. From an organizational perspective, if we have the knowledge of all the participating agents' preferences in the organization, the overall evaluation of the agents' interactions can be elaborated. However, in domains where privacy issues are concerned, some agents might not want to disclose their preferences. In such cases, mechanisms are needed to deal with incomplete information in the combined reasoning of agent interactions at the organization level.

An assumption of applying the combined reasoning approach is that the preferences of the actors can be obtained. It is either the case that the actors themselves have the knowledge of such information, or it is possible to derive such information from the past behavior of the actors. In the latter case, there is a separate research domain where a number of approaches have been proposed for mining and eliciting the preferences of actors (e.g., [107], [125]). Furthermore, we have shown that actors usually have some primitive preferences which are at a high level of abstraction and rarely change over time. When they participate in different organizational contexts, their primitive preferences need to be elaborated such that they can reason about the context-specific occurrences.

Part II

Application Studies

Chapter 8

Application: Social Structure

This Chapter has the following contributions:

- two application studies of social structure: a railway system and an international trade system,
- some design guidelines for the construction of social structures

In this Chapter, we present two application cases for the illustration of how the social structure of an organizational model is constructed. The first case concerns a train maintenance system [121], by which we show how social structures can be derived from the operational model of an organization. The second case study concerns an international trade system [120], by which we show how social structures can be derived by starting with the general objectives of an organization. Based on the two application cases, finally we provide some design guidelines for the construction of social structures in organizational design.

8.1 Train Maintenance system

8.1.1 Case Description

Railways are made up of hundreds of thousands of moving parts. If a railway service is to be reliable, the equipment must be kept in good working order and

Parts of this chapter have been published. Section 8.1 is based on our contribution to the 2012 IEEE/WIC/ACM International Conference on Intelligent Agent Technology, pp. 196-203, 2012 [121]; Section 8.2 is based on our contribution to the International IFIP Electronic Government Conference, pp. 308-319, 2011 [120]; Section 8.3 is based on our contribution to the 13th International Workshop on Coordination, Organisations, Institutions and Norms @ IAT 2011, pp. 58-74, 2011 [119].

regular maintenance is the essential ingredient to achieve this. In the railway system under our study, train maintenance currently occurs on a regular mileage or time basis. The question whether a job has to be executed, follows from inspection results obtained at the depot. To compensate for the uncertainty with respect to the execution of individual jobs, the volume of resource exceeds the average demand and slack is build into production plans. To improve the ratio of fleet availability and maintenance resources, as well as operational reliability, the maintenance department is planning to use information that is available from sensors and computers on board of trains, such that it is possible to model the actual status of trains and to predict orders for specific maintenance jobs.

Within the scope of the problem, different organization actors are involved, each having its own internal functional structure. We assume that each of them complies to some set of overall system objectives, so one can speak of an "Organizational Distributed Decision Making (DDM) System" [185]. In this DDM system, we consider three types of agents: human planners, software systems, and (parts of) organizations. Human planners make the final decisions, while the software systems support the process by communicating potential consequences of different plan options and by giving suggestions to solve conflicts and to optimize plans.

8.1.2 Current Situation Analysis

Operational Model

In the domain of train maintenance introduced in the previous section, the operational model of the current railway system is depicted in Figure 8.1, which describes the involved entities and their information dependencies. The three main entities involved are the Train Operating Company (TOC), the Infrastructure Manager (IM) and the Rolling Stock Maintenance company (RSMC). TOC makes the time schedule for the train services provided, while IM controls the physical flow of trains across the railway network. To enable the transfer of a specific train to one of the RSMC depots, TOC requests IM to assign the physical train to a path planning which finally leads to the depot. The time horizon spans approximately one week. RSMC prescribes to TOC both the mileage and time interval that is applicable to a specific train type, defines the content of the maintenance program (i.e. the set of jobs to be executed), and manages the depots. Within these entities, different organizations of sub-entities cooperate to realize the corresponding functionalities.

General Specification

At the top level, there are three major responsibilities of the system with respect to the three main entities involved, i.e., TOC, IM and RSMC. Accordingly, we can derive a general specification consisting of three interrelated roles Maintenance Order Provider (MOP), Transfer Service Provider (TSP) and Maintenance Service

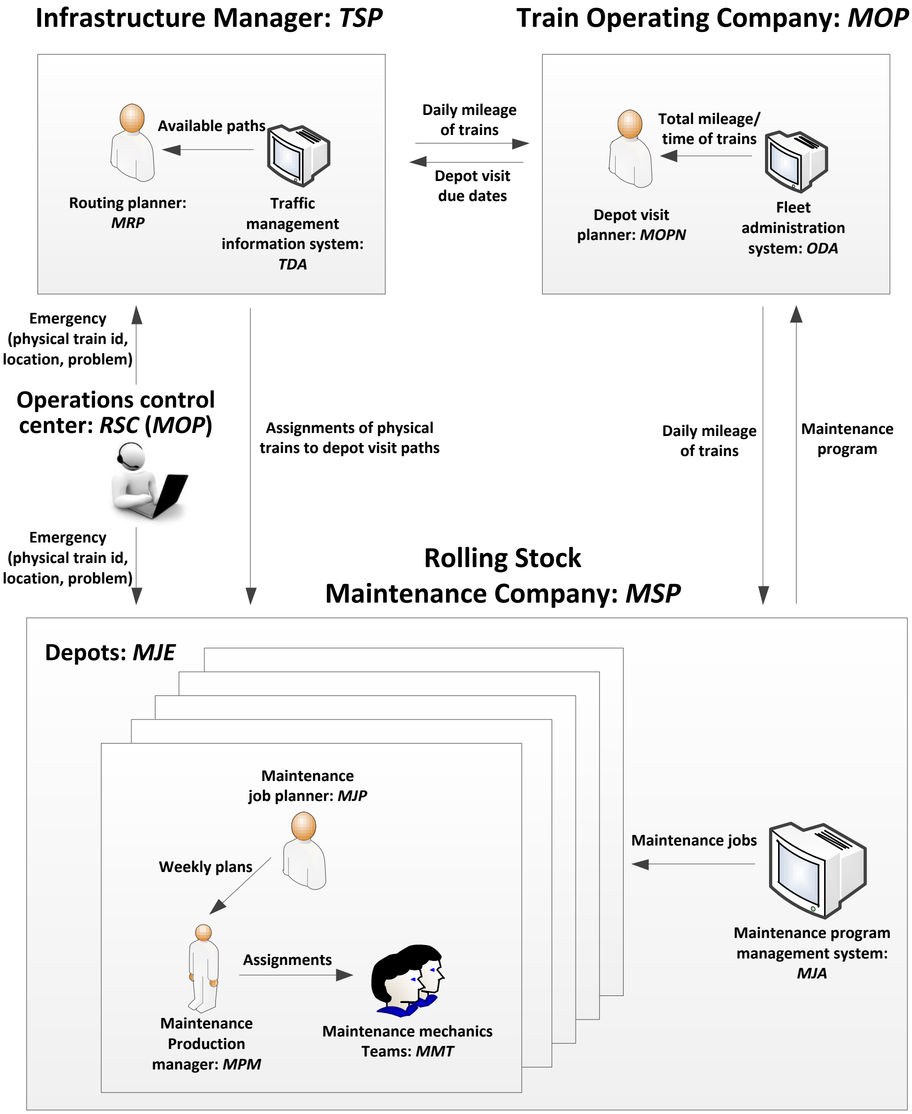


Figure 8.1: *Operational model of the current railway system*

Provider (MSP). MOP has the objective to plan maintenance orders according to the maintenance program or actual conditions. TSP has the objective to transfer trains to depots as requested by the MOP. MSP has the objective to execute maintenance tasks according to the maintenance program of the trains sent to the depots. MOP depends on TSP to transfer the train to depots when maintenance is needed, and depends on MSP to execute the maintenance tasks. MSP depends on TSP to transfer the train such that maintenance can be done. Formally, the general specification is defined as follows.

- SS^{gs}_{TM} = ({*MOP*, *TSP*, *MSP*}, {(*MOP*, *TSP*, *train_transferred*), (*MOP*, *MSP*, *train_maintained*(*train*, *report*)), (*MSP*, *TSP*, *train_transferred*(*train*, *route*))})

Contextual Specification

In this application case, we focus on a specific application domain concerning train maintenance. In this application domain, the main situational variable is the states of the trains, which are of two kinds, i.e., with the threshold usage of mileage and time, and with unexpected failure. Corresponding to these two states, the railway system delivers two types of services. One is regular maintenance by planned depot visits. The other is emergent repair by unplanned depot visits. Given the two kinds of states of the trains and the corresponding services delivered by the railway system, two sub-domains can be identified, captured by two contexts. The first context ctx_{PM} is called *planned maintenance* which corresponds to the regular maintenance service for the trains with the threshold usage of mileage and time. The second context ctx_{UM} is called *unplanned maintenance* which corresponds to the emergent repair service for the trains with unexpected failure. Focusing on different situations, these two contexts pose differing requirements for the enactment of the roles presented in the general specification SS^{gs}_{TM}, which derives two contextual specifications.

1. **Planned maintenance**

According to the maintenance program, trains are sent to depots for inspection and repair at regular intervals. TOC is responsible for checking the mileage and time usage of each train and delivering them to a depot according to the maintenance rules. Usually, the depot visit order is planned one week in advance so that IM can plan a physical route to transfer the train to a depot. At the same time, the depot will receive related information from IM about the expected arrival time of the train and from TOC the usage data of the train. When the train arrives at the depot, there are mainly two categories of maintenance work that will be done: (a) ordinary inspection and repair maintenance, and (b) additional work such as replacing main parts, small design changes and overhaul work, which usually cost more time and resources.

In this context, MOP enables an organization of two sub-roles: Operational Data Administrator (ODA), Maintenance Order Planner (MOPN). ODA has the objective to signal the need of specific trains for planned maintenance. It should have the capability to process the usage data delivered daily by TSP, to compare the cumulative data with the applicable thresholds, and to report on the remaining gaps. To guarantee safety, MOPN has the objective to let the trains be maintained before the critical mileage or time interval is exceeded and to assure fleet availability, while minimizing the life-cycle cost, and it depends on ODA for the mileage and time usage of the trains. MOPN requires the capabilities to generate, update, and communicate feasible depot visit plans, and to maximize the use of the trains based on the mileage and time interval allowed.

TSP consists of two sub-roles: Traffic Data Administrator (TDA) and Maintenance Routing Planner (MRP). TDA has the objective to provide information about the availability of routing options, to keep track of the assignment of trains to planning paths, and to provide daily mileage data of trains. It should have the capability to store, update and query routing plans over time. MRP aims to facilitate the transfer of trains to depots as requested by MOP by making feasible routing plans for specific trains and by managing the plan execution, and it depends on TDA for the availability of routing options.

MSP is enabled by a lower-level organization of the two sub-roles Maintenance Job Administrator (MJA) and Maintenance Job Executer (MJE). MJA has the objective to keep track of the maintenance jobs performed on trains and to signal the need for new jobs. Since MOP acts upon the usage data or urgent failures with the aim to have a train merely transferred to a depot, MJA has to identify the precise content of the work to be done, i.e. it has to specify the specific jobs to be scheduled. It requires the capabilities to keep track of the physical configuration of trains and the usage of the components, and to generate jobs regarding planned maintenance, component exchange, and modifications. MJE itself can be decomposed into three sub-roles, Maintenance Job Planner (MJP), Maintenance Production Manager (MPM), and Maintenance Mechanics Team (MMT). MJP has the objective to assign feasible working plans to train depot visits. It should have the capability to group the jobs generated by MJA in such a way that the plan meets the time constraints and resource availability. MPM has the objective to assign jobs for execution to MMT, monitor the progress, and adjust the plan when necessary. It requires production management capabilities. Finally, MMT has the objective to actually perform the maintenance jobs as assigned to them, according to the work descriptions and safety regulations. Agents enacting this role should be trained and certified to do so. In the organization of MSP, MJA depends on MJE to have the maintenance jobs executed. In the organization of MJE, MJP depends on MPM to have the jobs assigned and depends on MMT to have the jobs performed.

Based on the description above, we then derive a contextual specification from the general specification SS^{gs}_{TM} with respect to ctx_{PM}, formalized as below.

- $ctx_{\text{PM}} = \{regular\_maintenance\_service\}$,
- $SS^{cs}_{\text{TM1}} = (\{MOP, TSP, MSP\}, \{(MOP, TSP, train\_transferred), (MOP, MSP, train\_maintained(train, report)), (MSP, TSP, train\_transferred(train, route)))$ where

– $org_{\text{MOP}} = (\{ODA, MOPN\}, \{(MOPN, ODA, usage\_obtained(train, usage))\})$,
– $org_{\text{TSP}} = (\{TDA, MRP\}, \{(MRP, TDA, routing\_done(train, route))\})$,
– $org_{\text{MSP}} = (\{MJA, MJE\}, \{(MJA, MJE, maintenance\_done(train, report))\})$,
– $org_{\text{MJE}} = (\{MJP, MPM, MMT\}, \{(MJP, MPM, job\_assigned(train, job)), (MJP, MMT, job\_performed(train, job))\})$.

2. **Unplanned maintenance**

When unexpected failures occur, intermediate depot visits are needed. IM has to plan an emergent route so that the malfunctioned train can be transferred to a nearby depot. When the train is assigned to a depot, the depot has to make a quick response and adapt its maintenance plan. In order to coordinate the actions towards this unexpected train failure, the Operations Control Center (OCC) is involved to communicate the emergent problems of the train to IM and RSMC.

The elaboration of the two roles TSP and MSP in the unplanned maintenance context is the same as that of the planned maintenance context. As for MOP, a lower-level organization named Repair Service Coordinator (RSC) is introduced in the unplanned maintenance context. RSC has the objective to have trains repaired when their condition requires immediate action due to unexpected failure. It should be capable of identifying the services required, to request for a suitable routing, and to tune the schedule with the depot involved. As a result, we derive another contextual specification from the general specification SS^{gs} with respect to the context ctx_{UM} formalized as below.

- $ctx_{\text{UM}} = \{emergent\_repair\_service\}$,
- $SS^{cs}_{\text{TM2}} = (\{MOP, TSP, MSP\}, \{(MOP, TSP, train\_transferred), (MOP, MSP, train\_maintained(train, report)), (MSP, TSP, train\_transferred(train, route)))$ where

  – $org_{\text{MOP}} = (\{RSC\})$,
  – $org_{\text{TSP}} = (\{TDA, MRP\}, \{(MRP, TDA, routing\_done(train, route))\})$,
  – $org_{\text{MSP}} = (\{MJA, MJE\}, \{(MJA, MJE, maintenance\_done(train, report))\})$,
  – $org_{\text{MJE}} = (\{MJP, MPM, MMT\}, \{(MJP, MPM, job\_assigned(train, job)), (MJP, MMT, job\_performed(train, job))\})$.

From the analysis above, we can see that the railway system in the domain of train maintenance is a result of multi-organizational interactions at multiple levels of abstraction. Figure 8.2 gives a graphical illustration of the general and contextual specifications of the social structure. At an abstract level, three interrelated roles, i.e., MOP, TSP and MSP, build up an organization that achieves the overall objectives of the system. Given the refinement of the general train maintenance domain, the roles at the abstract level are extended with finer-grained components that collaborate in such a way that the objectives of the roles at higher levels can be achieved in a specific context.

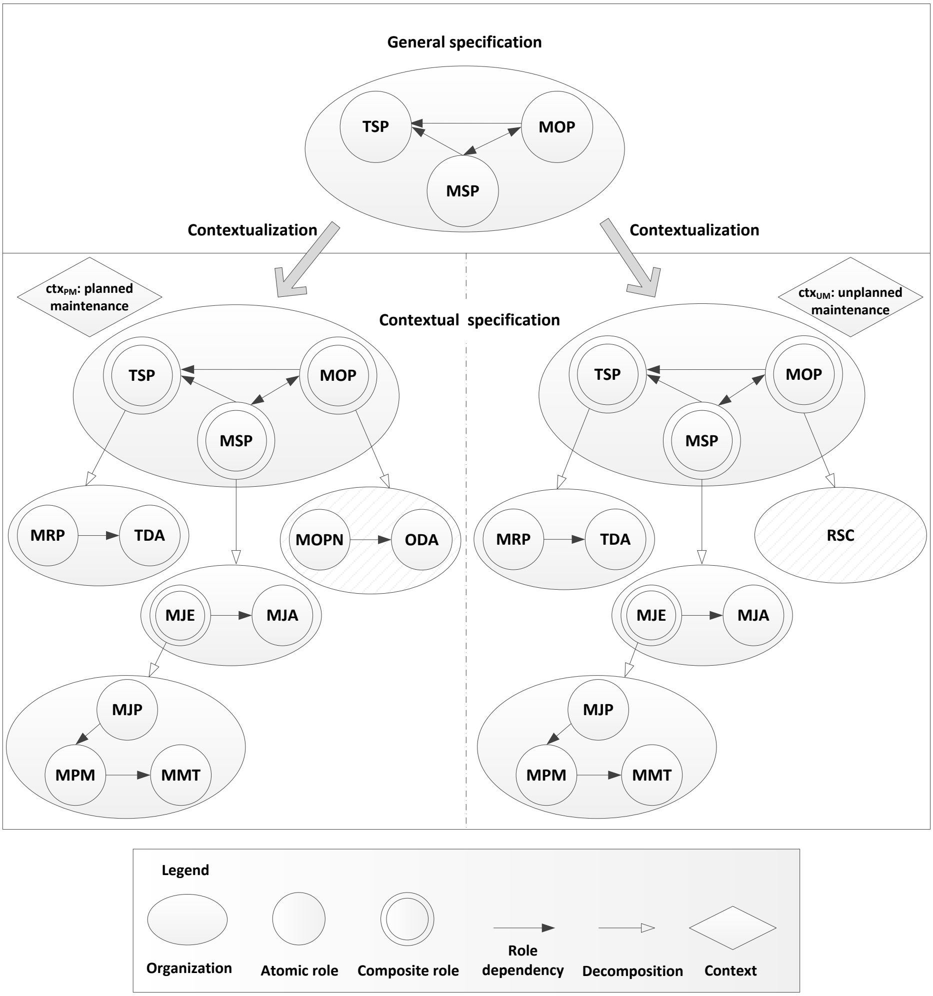


Figure 8.2: *Specification of the social structure of the current railway system*

8.1.3 Future Design

Introducing Artificial Agents

With the development of artificial intelligence, not only can intelligent agents accomplish advanced functionalities but also assist decision making by their sophisticated communicating ability. Table 8.1 shows the detailed information of the artificial agents introduced in the new system design.

Table 8.1: *Artificial agents introduced in the new system design*

Artificial agents	Capabilities
Train Sensor (TS)	Detects or measures different status conditions of trains
Train Agent (TA)	Status analysis, depot visit planning, social behavior
Maintenance Routing Agent (MRA)	Automatic scheduling of depot visit path, social behavior
Maintenance Planning Agent (MPA)	Automatic scheduling of maintenance plan, social behavior
Depot Coordinating Agent (DCA)	Negotiating, conflict solving

Each train will be installed with intelligent sensors (Train Sensor, TS) that can capture real-time status information of the train such as mileage, time, temperature, pressure, etc. With such information, the Train Agent (TA) that attached to the train can decide whether the train needs maintenance service or not. Moreover, TA is capable of social behavior so that it can communicate with other agents to improve their performance. For example, TA may adjust its depot visit plan according to the observation of the workload of different depots.

Maintenance Routing Agent (MRA) and Maintenance Planning Agent (MPA) are respectively responsible for assisting human beings in making decisions on transfer paths and maintenance plans. To achieve these, MRA and MPA will communicate with TA to know their depot visit requirements and resource needs of the trains so that TSP and MSP can obtain a routing-feasible and resource-feasible schedule optimized in some sense.

Depot visit orders are distributed to several depots in the maintenance company. Since the resources of the depots are limited and their expertise is different, there might be conflicts when they receive maintenance orders from train agents. At this time, Depot Coordinating Agent (DCA) will be in charge of solving such conflicts by coordinating between MPAs and making optional solutions.

With the artificial agents, on the one hand, real-time information can be obtained and distributed through the whole system. On the other hand, interactions between human decision makers and artificial agents improve the communication between different organizations. Figure 8.3 shows the operational model of the future design in which interactions between human and artificial agents play an important role. It can be seen that train agents collect all the status information about trains and communicate to IM and RSMC about their depot visit requests and resource needs. Plan negotiation is finalized by a routing and job commitment.

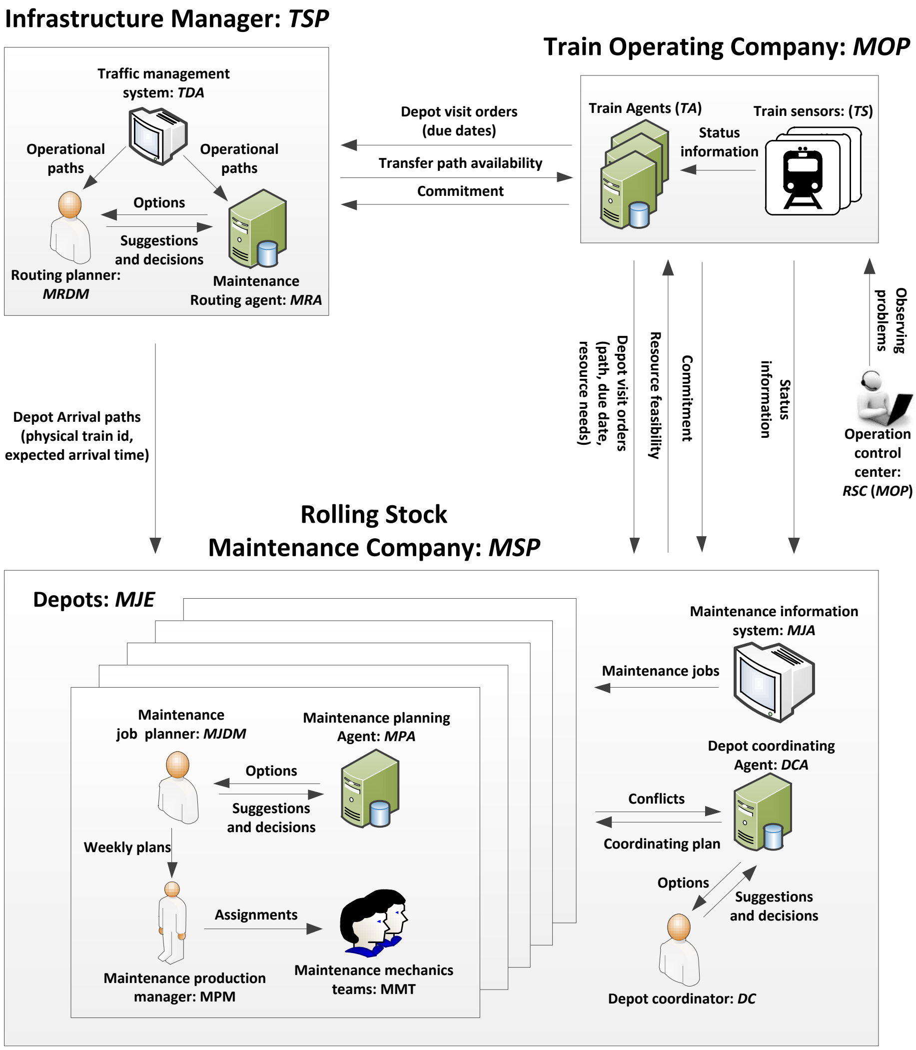


Figure 8.3: *Operational model of the future railway system design*

Context Transition

In the new system, due to the real-time monitoring ability of train sensors, and the advanced planning, communicating ability of train agents, depot visits of trains will be planned based on real needs and thus prevent unexpected failures to a large extent. Accordingly, a new context called *adaptive maintenance* marked as ctx_{AM} emerges in the new system. It not only incorporates a part of the unplanned maintenance domain but also extends the planned maintenance domain, since more status information of trains can be achieved and therefore spreads the service of train maintenance to a wider domain. In this new context, train maintenance is based on real-time status information of trains from two sources. One is from intelligent train sensors and the other one is from the operational control center who collects emergent failure of trains and serves as a manual sensor.

In the new system, train maintenance will mainly be dealt with in the adaptive maintenance context which not only reduces the uncertainty of maintenance plans but also improves the flexibility of the scheduling task. However, the other two contexts still remain in the new design so that the railway system can gradually adapt to the changes.

Model Evolution

Given the context transition, the general specification of the social structure derives another contextual specification with respect to the adaptive maintenance context in the new design as shown in Figure 8.4. It can be seen that in this contextual specification more specific roles are added with the introduction of artificial agents. In MOP, TA takes the responsibility of planning maintenance orders based on the status information received from TS or RSC. Another important evolution is that the objective of MRP and MJP is distributed to two kinds of roles, i.e., Maintenance Routing Decision Maker (MRDM), Maintenance Job Decision Maker (MJDM) who make the final decisions, and MRA, MPA who support the process by communicating potential consequences of different plan options and by giving suggestions to solve conflicts and optimize plans. Besides, a role of Depot Coordinator (DC) is added in MSP so that conflicts between the actors of MJE can be managed with the help of DCA.

8.1.4 Reflection

In this application case, we explained the process of constructing a social structure based on the operational model of a railway system, and presented an analysis on the organizational evolution of the railway system from the perspective of train maintenance. It has been shown that organizational interactions in both the current real-life system and a potential future design are systematically described. Comparing with the specification of the current railway system, we can clearly see the evolution of the new organizational design in terms of the roles, dependencies and situational domains where interdependent roles are situated to achieve the

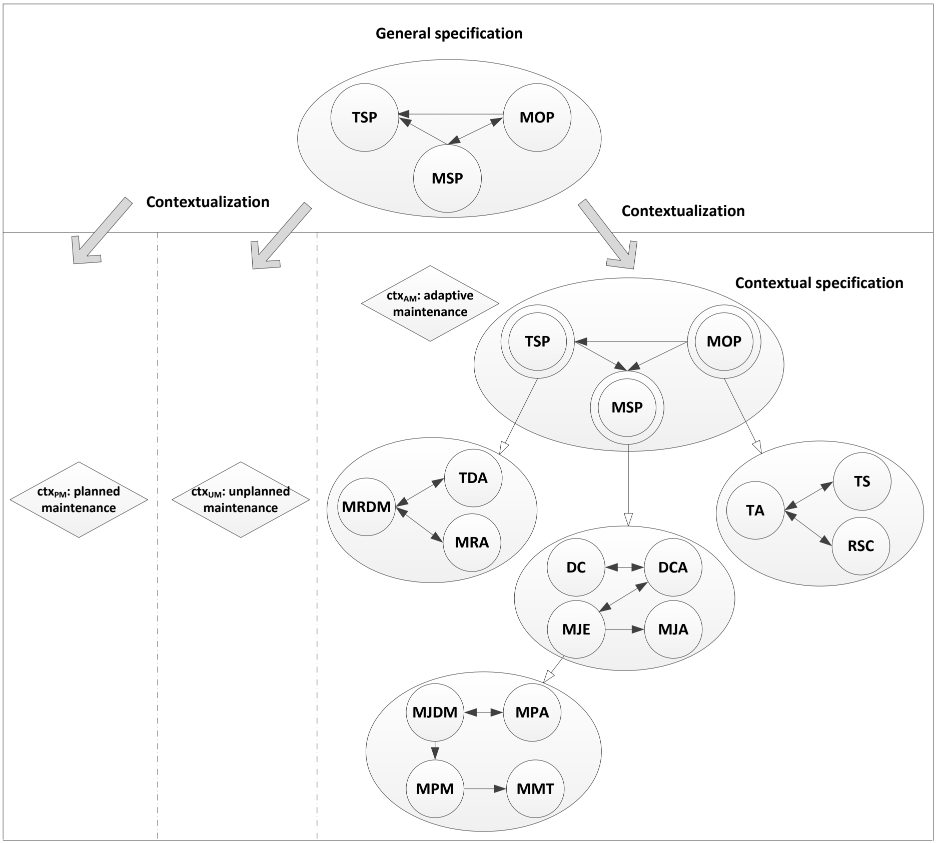


Figure 8.4: *Social structure of the future railway system design*

overall objective. From the decision making point of view, this model supports discussion and communication between people with different background knowledge, since it provides a comprehensive analysis of organizational interactions from abstract values to concrete operations with respect to different contexts. On the other hand, with a structured description linked to operational agents, the model provides a basis for an implementation of the new system design.

8.2 International Trade System

8.2.1 Case Description

With the growth of international trades and increasing security threats, governmental authorities are confronted with increasing regulation expenses while business companies are faced with increasing transaction time and costs. To these ends, interactions between business and governmental organizations are changing from monolithic control by regulatory authorities to distributed environments where private enterprises are responsible to regulate their affairs.

AEO (Authorized Economic Operator) program [59] is an EU effort that aims to secure international trades while at the same time reduce the administrative burden for companies through the use of self-regulation. Customs authorities grant AEO certificates to companies that have the capability to guarantee the effectiveness of their control systems, their economic stability, and to observe existing customs rules. Certification will result in an extensive reduction in the frequency of both document-based and physical inspections.

With the introduction of AEO certificates, the regulative relations between business and governmental organizations are differentiated. In order to facilitate smooth and secure business operations, there is a strong need for organizations to understand their responsibilities and opportunities according to the changing environments.

8.2.2 General Specification

At an abstract level, international trade systems have the objective of secure and efficient goods and service exchanges. Given that our focus is on the effects of regulative relations, we generalize an organization of two roles for the international trade system at this level, i.e., regulatory authorities (RA) and private enterprise (PE). RA and PE have very different objectives but they are complementary in accomplishing the overall objective of the international trade system, i.e., RA aims at ensuring the safety of the business operations by efficient regulation while PE tries to accelerate the trading process by efficient business operations. To realize their individual objectives, RA relies on PE for assisting in regulation-related tasks while PE depends on RA for operation permissions and a reliable business environment. As a result, we can derive a general specification of the social structure of the international trade system, formalized as follows.

- $SS^{gs}_{\mathrm{IT}} = (\{RA,\ PE\},\ \{(RA,\ PE,\ \mathit{regulation\_done}(\mathit{transaction},\ \mathit{report})),\ (PE,\ RA,\ \mathit{operation\_permitted}(\mathit{transaction}))\})$

8.2.3 Contextual Specification

With the introduction of AEO program, the relations between the governmental authorities and private enterprizes are of two kinds [38]. One is that governmental organizations do not trust those business companies without AEO certificates and directly take controls over the companies' operations. The other is that business companies with AEO certificates are trusted by governmental organizations for self control. In this case, the AEO program serves as a situational variable which influences the state of the relation between business and governmental organizations. Thus, corresponding to the two kinds of relations resulting from the AEO program, we abstract two contexts respectively named direct control (ctx_{D}) and self regulation (ctx_{S}), in which the objectives of the roles presented in the general specification are achieved in different ways.

According to the specific requirements of the two contexts, we obtain two different contextual specifications, illustrated as follows.

1. **Direct control:**

This context ctx_{D} concerns the situations where the private enterprises do not meet the self-regulation standards set by the governmental authorities and thus the governmental authorities have to take the responsibilities of the tasks relating to business regulations. Accordingly, only RA is elaborated into a composite role that refers to an organization of five sub-roles: Norm Specifier (NS), Control Indicator Maker (CIM), Action Monitor (AM) and Sanctioner (SC). NS has the objective to specify an effective set of norms for the regulation of the private enterprizes, and it requires the capabilities of mastering business domain knowledge and norm formulation techniques. CIM aims at determining the appropriate control indicators for the regulation of the private enterprizes, and it requires the capabilities of recognizing the regulation purposes of the norms and abstracting such purposes. In order to realize its objective, CIM depends on NS to have the norms specified from which the control indicators are derived. A control indicator is the kind of evidence required to demonstrate compliance of a norm, as well as infrastructural requirements to collect that evidence. AM has the objective to effectively monitor the actions of the private enterprizes, and it requires the capabilities of setting up monitoring system and retrieving useful data from the monitors. AM depends on CIM to have the control indicators made such that it can carry out the monitoring tasks with respect to the information about the control indicators. SC has the objective to impose the right sanctions on the private enterprizes in case that violations are detected, and it requires the capabilities of recognizing violations and executing sanctions. SC depends on AM for the accurate monitoring results such that the information about violations can be achieved to trigger the imposition of sanctions. Based on the descriptions of the five sub-roles and their interdependencies, we can derive a contextual specification from the general specification SS_{IT}^{gs} with respect to the context ctx_{D}, formalized as follows.

- $ctx_{\text{D}} = \{without\_AEO\}$,
- $SS_{\text{IT}}^{cs1} = (\{RA, PE\}, \{(RA, PE, regulation\_done(transaction, report)), (PE, RA, operation\_permitted(transaction))\})$ where
  - $org_{\text{RA}} = (\{NS, CIM, AM, SC\}, \{(CIM, NS, norms\_specified(norms)), (AM, CIM, indicator\_made(indicators)), (SC, AM, monitor\_done(violations))\})$.

2. **Self regulation:**

This context ctx_{S} concerns the situations where the private enterprises are granted AEO certificates and thus are trusted by the governmental authorities to take a part of the regulation responsibilities from the regulative authorities. Accordingly, both RA and PE transform to a composite role, and some of the roles in the lower-level organization of RA in ctx_{D} shift to the lower-level organization of PE in ctx_{S}. As for PE, it enables an organization of three sub-roles: Norm Specifier (NS), Control Indicator Maker (CIM) and Action Monitor (AM). These three roles have the same objectives and dependencies as that in the context of direct control. In this way, the control and monitoring responsibilities are transfered from the regulative authorities to the private enterprizes, and thus relieve the burden of the regulative authorities. As for RA, it enables an organization of two sub-roles: Control Monitor (CM) and Sanctioner (SC). Instead of AM, CM is specified in RA, which aims at monitoring the control activities of PE rather than the general activities of PE, and it requires the capabilities of communicating with the monitor system of the private enterprizes and retrieving the information relating to the control indicators. SC has the same objectives as that in the context of direct control, but now depends on CM for the accurate monitoring results such that the information about violations can be achieved to trigger the imposition of sanctions. Based on the descriptions of the sub-roles and their interdependencies with respect to the organization of RA and PE, we derive another contextual specification from the general specification SS_{IT}^{gs} with respect to the context ctx_{S}, formalized as follows.

- $ctx_{\mathrm{S}} = \{with\_AEO\}$,
- $SS_{\mathrm{IT}}^{cs2} = (\{RA, PE\}, \{(RA, PE, regulation\_done(transaction, report)), (PE, RA, operation\_permitted(transaction))\})$ where
  - $org_{\mathrm{RA}} = (CM, SC\}, \{(SC, CM, monitor\_done(violations))\})$,
  - $org_{\mathrm{PE}} = (\{NS, CIM, AM\}, \{(CIM, NS, norm\_specified(norms)), (AM, CIM, indicator\_made(indicators))\})$.

According to the general and contextual specifications described above, Figure 8.5 gives a graphical representation. Starting with the general objectives of the system, two atomic roles RA and PE are identified at the top level, indicating the social relations between regulative authorities and private enterprises from an abstract view. Then according to the two refined contexts determined by whether an agent enacting the role PE is granted a AEO certificate, the two roles are refined into different organizations of sub-roles, imposing different requirements for accomplishing the general objectives of the system.

8.2.4 Reflection

In this application case, we explained the process of constructing a social structure starting with some general objectives of an international trade system, and

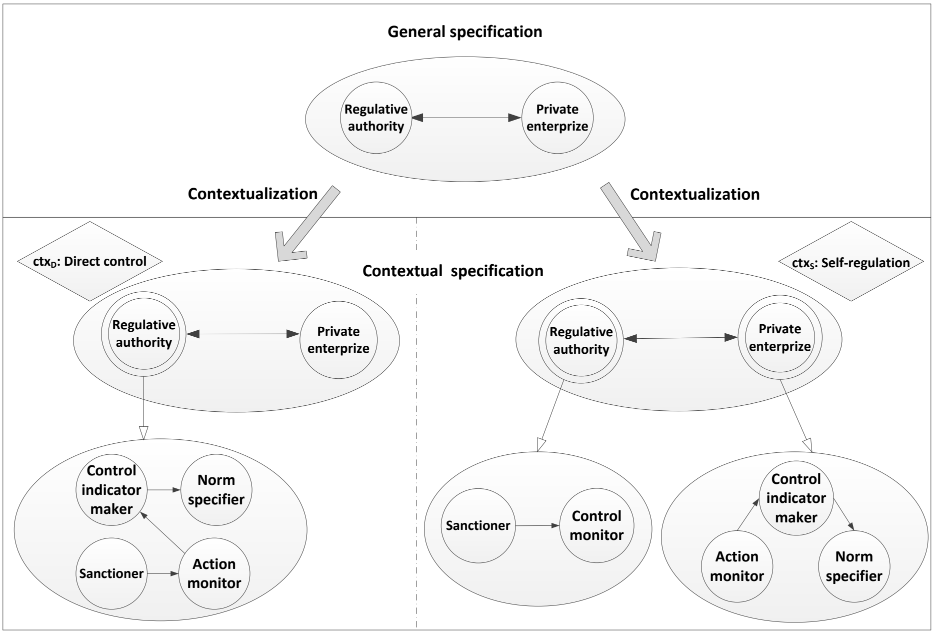


Figure 8.5: *Social structure of the international trade system*

presented an analysis of the changes of requirements with respect to the social relations between regulative authorities and private enterprizes. In practice, there are two difficulties with respect to the interactions between business and governmental organizations: (1) business organizations have to elicitate themselves the norms from the governmental organizations, (2) the norms have to be customized to the specifics of each business organization, e.g., safety risks for a dairy company is primarily food-safety, whereas safety risks for a scrap metal trading company is hazardous waste, or even hidden bombs. To this end, the social structure of OperA+ can help to solve these two difficulties: (1) by structuring the contextual refinement modeling process to guide both business organizations and governmental organizations to build their interactions from an abstract level to a concrete level, which provides both of them a better understanding of their responsibilities, and (2) by using contexts to differentiate the interaction between business organizations and governmental organizations according to their status, i.e., business type, regulation policy, etc.

The two application studies above show, from different aspects, the construction of social structures of organizational models and the analysis of organizational interactions. In the next section, we present some design guidelines drawn from our experience obtained from the application studies.

8.3 Design Guidelines

Following a modular way of modeling and analyzing social relations, the social structure of an OperA+ model subdivides a system into lower-level organizations of roles that can be independently designed and then re-used in different systems to drive multiple functionalities. Based on the compositionality of roles and agents in OperA+, we divide the whole modeling process into four phases as shown in Figure 8.6. The vertical dimension is from abstract contexts to specific contexts while the horizontal dimension is from roles to agents.

In abstract contexts, organizational interactions are described by roles with very general objectives so that agents enacting those roles have much autonomy but little guidance and constraints on how to enact those roles. To ensure that the objectives of the roles can be accomplished, on the one hand, roles in abstract contexts will be elaborated into more finner-grained components with respect to more specific contexts, and on the other hand, eligible agents will be employed to join in the organization and enact the roles. This results in a complete picture of the social structure. Such a way of constructing social structures can be reflected from the second application case presented in Section 8.2. Another way of constructing social structures can be found in the first application case presented in Section 8.1. Initially, we have the operational model of a system, by which we know the agents employed by an organization in various contexts. With respect to a specific context, the division of roles can be derived from the tasks and functionalities of the agents in the system.

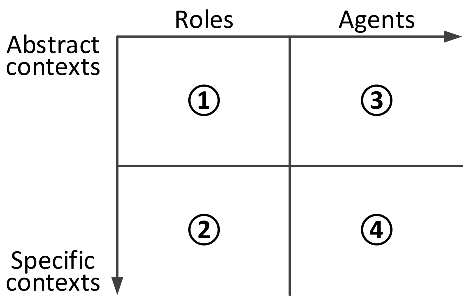


Figure 8.6: *Modeling process*

8.3.1 Roles in abstract contexts.

For a system to be modeled, there must be some initial objectives that trigger the system. These objectives are always described at an abstract level and only give the general ideas about what to do. The roles in this phase can be derived from related applications according to the initial requirements of the model. Therefore, they can be reused by other organization models with similar requirements. In practice, organizational models are of different types such as inter-organizational projects, supply chains. These types of collaboration models have their own universal set of basic roles which can be seen as templates. Based on the nature of an organizational collaboration, we can inherit the basic roles from the related templates. That is, we have some patterns to use in the first phase. Specifically, some organizational collaboration models may cover multiple types and they can inherit the basic roles from all the templates of these types. For role dependencies, a similar mechanism can be used.

8.3.2 Roles in specific contexts.

Based on the roles derived from abstract contexts, further elaboration is needed for specific applications. In this phase, the roles are case specific and are obtained based on the characteristics of specific collaboration requirements. Organizational collaboration models with the same set of roles in an abstract context may have different requirements in specific contexts and result in different lower-level organizations of sub-roles. That is, with the refinement of contexts, an organizational collaboration model at an abstract level is contextualized into finer-grained organizations of components with respect to different application environments. With more specific objectives, agents can know what is actually needed to be done in a specific situation. However, "specific" is in a relative sense and the degree of specificity should balance between system control and actor autonomy. Therefore, on the one hand, enough regulations are imposed on the agents' behavior so that the implementation of the system does not deviate from the original goals. On the other hand, the agents have enough autonomy to decide their participation and collaborating strategies.

8.3.3 Agents in abstract contexts.

In accordance with the roles in abstract contexts, potential agents can be identified. Generally speaking, no single agent can accomplish the objectives of a role in an abstract context. This is due to the fact that organizations in real business world usually focus on a few business fields and have limited resources. No single organization can do all the work in a large business process that involves several business fields and requires a large quantity of resources. Therefore, we should construct a pool of potential enactors, i.e. candidate agents. All the qualified agents can be added into the pool for further selection. In this phase, we only know the general objectives from the roles in abstract contexts, and these objectives serve as the basic requirements for the agents who want to participate in an organizational collaboration.

8.3.4 Agents in specific contexts.

In the last phase, we should figure out the specific agents for the roles in a specific context. Then the objectives of the roles can be accomplished as expected. For this, the second phase and the third phase which describe the different aspects of the collaboration model should be combined to derive the final results. According to the roles in a specific context, we know what objectives to achieve in a specific situation. From the agents in an abstract context, we select appropriate agents for the roles in specific contexts. Finally, we know exactly who achieves what objectives and the whole model is constructed.

8.4 Conclusions

In this chapter, we presented two application studies to evaluate the social structure of OperA+. The first case illustrates how a social structure of an organizational model can be derived from the operational model of an organization. The results show that such a process facilitates the understanding of organizational interactions that occur at multiple levels of abstraction and provides an evolutionary view on how organizational behavior is adapted to meet the specific requirements of changing contexts. The second case illustrates how a social structure of an organizational model can be derived from the general objectives of an organization. The resulting model gives a clear view on the differing requirements of the participants' behavior according to their situated contexts. Based on the practical applications, finally some design guides are provided to facilitate the construction of social structures in organizational design.

Chapter 9

Application: Normative Structure

This Chapter has the following contributions:

- an evaluation of the construction of normative structures and the compliance checking approach (presented in Chapter 5) based on two case studies from international trade domain,
- an evaluation of the modeling of collective institutions and the conflict detection approach (presented in Chapter 6) based on one case study from international trade domain, and

In this Chapter, we show three case studies based on the work presented in [115], [116], and [141], to evaluate the normative structure of OperA+. In the first case study, we focus on the modeling of a set of regulations adopted from the domain of international trade, and show how compliance diagnosis may help in deriving compliant interaction processes. In the second case study, we show how a business process designed by an organization can be verified against a set of legal regulations. The third case study concerns the modeling of collective institutions and detection of norm conflicts.

Parts of this chapter have been published. Section 9.1 is based on our contribution to the Journal of ACM Transactions on Management Information Systems, to be published, [115]; Section 9.2 is based on our contribution to the Journal of AI and Society, pp. 1-11, 2014 [116]; Section 9.3 is based on our contribution to the 13th International Workshop on Coordination, Organisations, Institutions and Norms @ AAMAS, pp. 136-154, 2013 [141].

9.1 Entry Summary Declaration

9.1.1 Case Description

As the first case study, we consider the process of entry summary declaration in European Union (EU), which is concerned with trade security and efficiency. For example, it is important to assess the risk before the arrival of the goods and to ensure that no delays for the logistical chain occur. To this end, pre-arrival information is required to be made available in the form of an Entry Summary Declaration (ENS) before all goods are brought into the Community. Concerning the ENS, the following is a fragment of regulations formulated and enacted by the EU commission [62].

Article 183.1 The entry summary declaration shall be made electronically [*make_eENS*] [before the *Applicant* lodges the ENS (*lodge_ENS*)]. [. . .] The entry summary declaration shall be authenticated [*authenticate_ENS*] by the person making it [*Maker*] [before the applicant lodges the ENS].

Article 183.2 The customs authorities [*Customs*] shall allow the lodging of a paper-based entry summary declaration [*approve_pENS*], [. . .], only in one of the following circumstances:

- ***(a)*** the customs authorities' computerised system [*Computer_sys*] is not functioning [*malfunction*];
- ***(b)*** the electronic application of the person lodging the entry summary declaration [*Electronic_app*] is not functioning [*malfunction*].

[within 1 day (*pass_1day*) after the person making the ENS applies to lodge a paper-based ENS (*apply_pENS*)].

Article 183.4 The use of a paper-based entry summary declaration [. . .] shall be subject to the approval of the customs authorities [*approve_pENS*]. The paper-based entry summary declaration shall be signed by the person making it [*sign_pENS*].

Article 183.6 Entry summary declarations shall be registered [*register_ENS*] by the customs authorities immediately [*pass_1day*] upon their receipt [*receive_ENS*].

Article 184a.1 In the case of maritime traffic the entry summary declaration shall be lodged at the customs office of entry by the following deadlines:

- ***(a)*** for containerised cargo [*container_cargo*], [. . .], at least 24 hours [*pass_24hour*] before [*Carrier*] loading at the port of departure [*load*];
- ***(b)*** for bulk cargo [*bulk_cargo*], [. . .], at least four hours [*pass_4hour*] before [*Carrier*] arrival at the first port in the customs territory of the Community [*arrive*];

...

[otherwise, penalties in the national legislation shall be followed by the applicant (*follow_penalty*) within 1 week (*pass_1week*)].

Article 184d.2 The customs authorities shall complete the risk analysis [*complete_riskAnalysis*] prior to the arrival of the goods [*arrive*], [...].

It can be seen that when some goods are planned to be introduced into the EU customs territory, an ENS is required to be lodged to the customs authorities within a time limit. The ENS has to be made according to a number of constraints with respect to the format and the contents. After receiving the ENS, the customs authorities are responsible to perform risk analysis and determine whether the goods are allowed to be imported and whether any control measures need to be taken, before the goods arrive at the customs territory.

Given that actors cannot be fully controlled, it is necessary to facilitate continuous compliance checking of the business processes of both public and private organizations against the regulations. Moreover, since the formulation of such regulations is done manually, inconsistencies are likely, especially in dynamic environments. Thus, some important issues are raised. e.g., are the organizations complying with the regulations? Is it possible for them to achieve full compliance following their business processes?

9.1.2 Formalization

(1) *Regulative norms*

Given the regulations described above, we build a regulative norm net as follows. When introducing goods into the customs territory, an ENS has to be lodged to the customs. The ENS has to be made electronically and also be authenticated by the person making it, as specified by *Article 183.1* and formalized by RNN_1 = AND(n_1^r, n_2^r) where,

- n_1^r = (O, (*Maker*, *make_eENS*), (*Applicant*, *lodge_ENS*), (*Applicant*, *introduce_goods*))

[*Given that the Applicant introduces some goods into the customs territory (introduce_goods), the Maker is obliged to make an electronic ENS (make_eENS) before the Applicant lodges it to the customs (lodge_ENS).*]

- n_2^r = (O, (*Maker*, *authenticate_ENS*), (*Applicant*, *lodge_ENS*), (*Maker*, *make-_eENS*))

[*Given that the Maker makes an electronic ENS, the Maker is obliged to authenticate it (authenticate_ENS) before the Applicant lodges it to the customs.*]

However, there is an alternative to make a paper-based ENS given that certain conditions are satisfied, as specified by *Article 183.2* and formalized by n_3^r.

- n_3^r = (O, (*Customs*, *approve_pENS*), (*Applicant*, *apply_pENS*)<(*timeline*, *pass_1day*), (*Computer_sys*, *malfunction*)∨(*Eletronic_app*, *malfunction*))

[*Given that either the computer system of the customs (Computer_sys) or the electronic application of the applicant (Electronic_app) malfunctions, the Customs is obliged to approve the the use of a paper-based ENS (approve_pENS) within 1 day (pass_1day) after the Applicant applies for making a paper-based ENS (apply_pENS).*]

Once the customs authorities approve a paper-based ENS, the *Maker* has to follow the requirements for making a paper-based ENS, as specified by *Article 183.4* and formalized by RNN_2 = AND(n_4^r, n_5^r) where

- n_4^r = (O, (*Maker*, *make_pENS*), (*Applicant*, *lodge_ENS*), (*Customs*, *approve_pENS*))

[*Given that the Customs approves the use of a paper-based ENS (approve_pENS), the Maker is obliged to make a paper-based ENS (make_pENS) before the Applicant lodges the ENS (lodge_ENS).*]

- n_5^r = (O, (*Maker*, *sign_pENS*), (*Applicant*, *lodge_ENS*), (*Maker*, *make_pENS*))

[*Given that the Maker makes a paper-based ENS, the Maker is obliged to sign the ENS (sign_pENS) before the Applicant lodges the ENS.*]

RNN_2 and n_3^r have an AND relation and thus we get RNN_3 = AND(RNN_2, n_3^r). It can be seen that RNN_1 and RNN_3 specify two alternatives of making an ENS, i.e., electronic or paper-based. As long as an ENS is made following either of the two RNNs, the process is seen as compliant. In this sense, an OR relation can be assigned between RNN_1 and RNN_3, from which we get RNN_4 = OR(RNN_1, RNN_3).

After the *Customs* receives the ENS, they have to register the ENS, as specified by *Article 183.6* and formalized by n_6^r.

- n_6^r = (O, (*Customs*, *register_ENS*), (*timeline*, *pass_1day*), (*Customs*, *receive_ENS*))

[*Given that the Customs receives the ENS (receive_ENS), the Customs is obliged to register the ENS (register_ENS) within one day (pass_1day).*]

An AND relation between n_6^r and RNN_4 leads to $RNN_5 = AND(RNN_4, n_6^r)$. Moreover, the *Customs* has to complete the risk analysis based on the ENS lodged by the applicant before goods arrive, as specified by *Article 184d.2* and formalized by n_7^r.

- n_7^r = (O, (*Customs*, *complete_riskAnalysis*), (*Carrier*, *arrive*), (*Applicant*, *lodge_ENS*))

[*Given that the Applicant lodges the ENS, the Customs is obliged to complete the risk analysis (complete_riskAnalysis) before the Carrier arrives.*]

Similarly, we get RNN_6 = AND(RNN_5, n_7^r) by relating RNN_5 and n_7^r.

Importantly, there are time limits for lodging an ENS, as specified by *Article 184a.1* and formalized by RNN_7 = AND(n_8^r, n_9^r).

- n_8^r = (F, (*Carrier*, *load*), (*Applicant*, *lodge_ENS*)<(*timeline*, *pass_24hour*), (*Carrier*, *containerize_goods*))

[*Given that the Carrier containerizes the goods (containerize_goods), the Carrier is forbidden to load the goods 24 hours before the Applicant lodges the ENS.*]

- n_9^r = (F, (*Carrier*, *arrive*), (*Applicant*, *lodge_ENS*)<(*timeline*, *pass_4hour*), (*Carrier*, *bulk_goods*))

[*Given that the Carrier bulks the goods (bulk_goods), the Carrier is forbidden to arrive at the terminal 4 hours (pass_4hour) before the Applicant lodges the ENS.*]

In RNN_7 we use prohibitions to model the regulations in stead of obligations, following Von Wright's logic system [220]. If the regulations formalized in RNN_7 have not been followed, penalties from the national legislation will be applied. That is, if the carrier violates either of the two norms in RNN_7, the applicant has to accept the penalties specified by the national legislation within a time limit, as specified by *Article 184a.1* and formalized by n_{10}^r.

- n_{10}^r = (O, (*Applicant*, *follow_penalty*), (*timeline*, *pass_1week*), ∅)

[*The Applicant is obliged to follow the penalties from the national legislation (follow_penalty) within 1 week (pass_1week).*]

n_{10}^r serves as a sanction when RNN_7 is violated. Therefore, an OE relation can be assigned between RNN_7 and n_{10}^r, from which a norm net RNN_8 = OE(RNN_7, n_{10}^r) can be obtained. Furthermore, RNN_6 and RNN_8 should be followed simultaneously and finally a complete norm specification is built for the example, i.e., RNN_{ex1} = AND(RNN_6, RNN_8).

In RNN_{ex1}, there are 10 interrelated norms. The relation concerning shared components can be found, e.g., between n_1^r and n_2^r with the same deadline. For the relation concerning compliance status, e.g., RNN_1 and RNN_3 have an *OR* relation to indicate that either the person making the ENS (1) makes an electronic ENS and authenticate it, or (2) makes a paper-based ENS and sign it given that there is a failure of the computer system of the customs authorities or the electronic application of the person lodging the ENS, the collective behavior is seen as norm compliant.

Figures 9.1, 9.2 and 9.3 show the CPN model of RNN_{ex1} created in CPN tools[1]. Each place is represented as an ellipse with three pieces of information:

[1] http://cpntools.org/

a label (inside the ellipse), a color set (below the ellipse), and an initial marking (a black dot with a number inside). A transition is represented as a box with a label inside. An expression is defined on each arc connecting the places and transitions. Figure 9.4 presents the definition of the color sets and variables used in the CPN model of RNN_{ex1}. Agents are indicated by their names which are colored as *string*. Each role defines a color set with its name. In particular, we define a role named *Timeline* to represent time. Agents enacting a specific role is indicated by the product of the color of the role and the color of agents. The colors of complied and violated are unions of all the consisting roles enacted by agents. The type declarations for the color sets used in this case study are shown in Figure 9.4. Given the color sets, a list of variables are defined and colored accordingly.

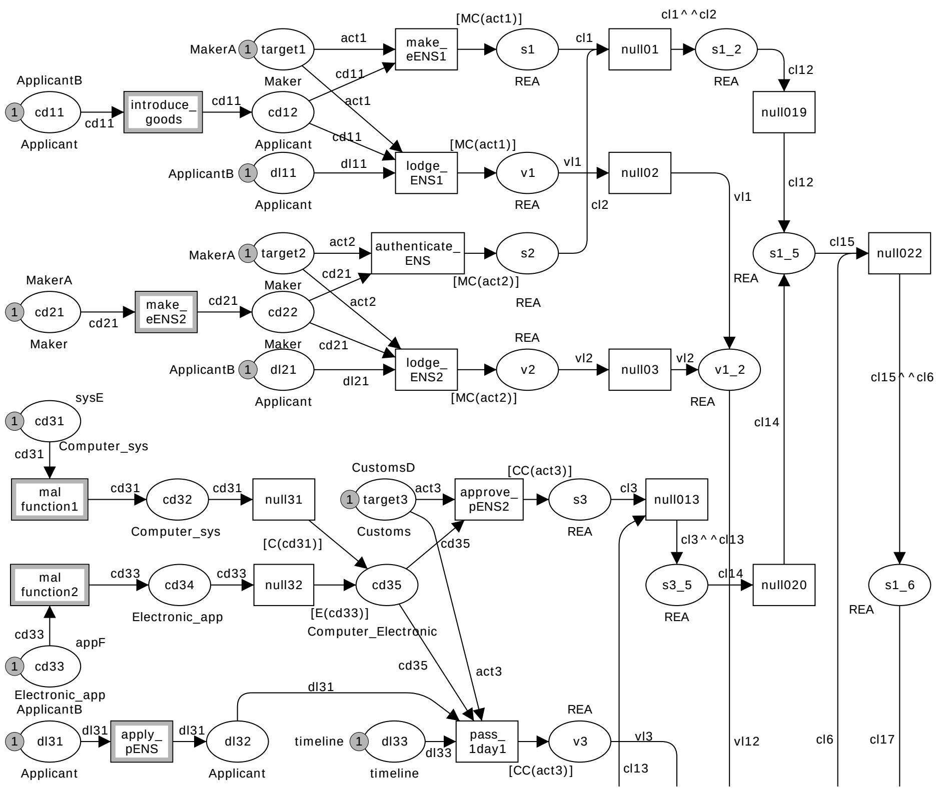


Figure 9.1: *The CPN model of RNN_{ex1} (continued in Figure 9.2)*

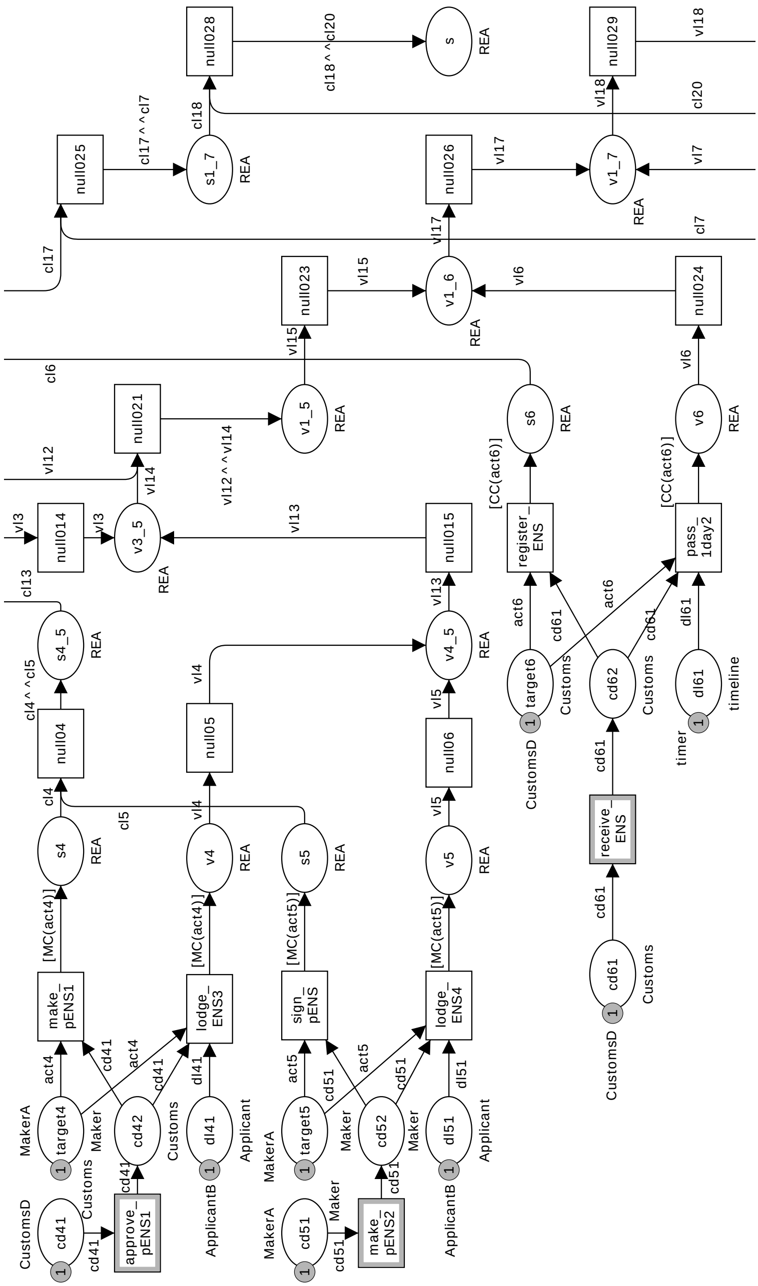


Figure 9.2: *The CPN model of RNN_{ex1} (continued in Figure 9.3)*

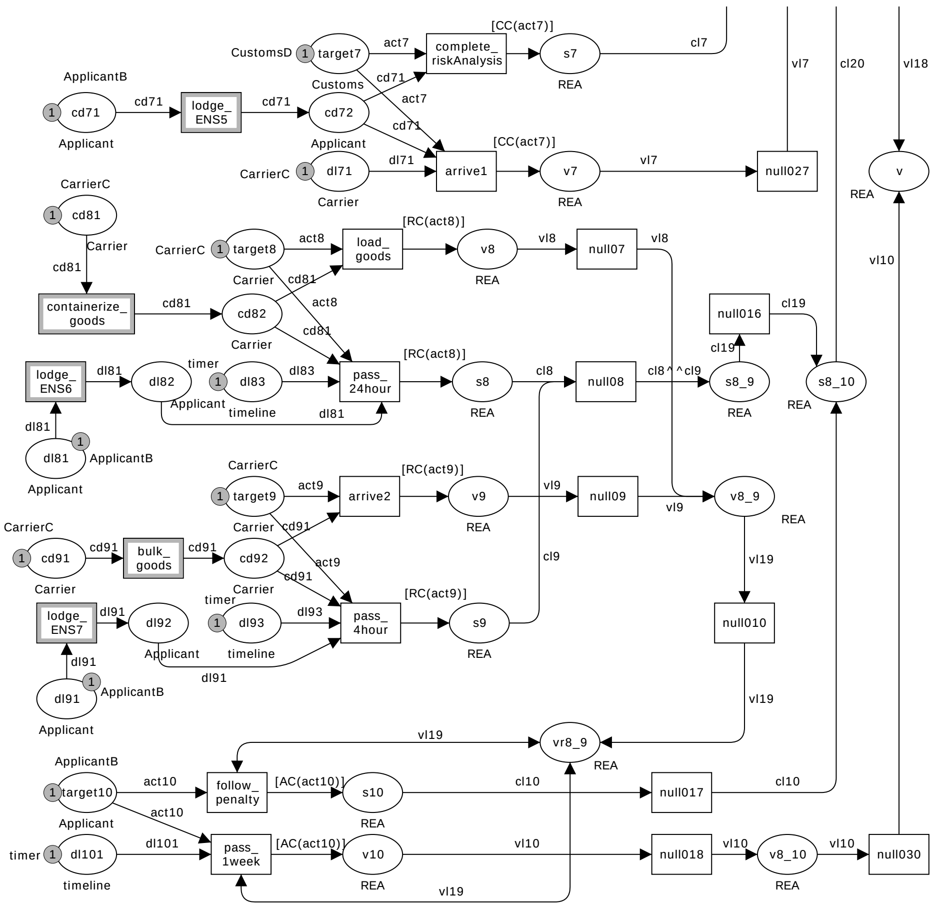


Figure 9.3: *The CPN model of RNN_{ex1} (continued from Figure 9.2)*

There are 10 norms in RNN_{ex1} and their CPN constructions can be distinguished by the labels of the places and arc expressions. For example, in the place labeled *cd*11, *cd* indicates that the place belongs to the CPN construction of the precondition of a norm, the first "1" indicates that it is the first norm n_1^r in RNN_{ex1}, and the second "1" is simply a sequence number. Similarly, *dl* indicates the deadline of a norm in the CPN construction. Specifically, there are two sink places in this CPN model and they are respectively labeled c and v, indicating the overall complied and violated state of NN_{ex2}. With one OE relation in NN_{ex2}, there is accordingly one place labeled vr, indicating the repairable violation state. Moreover, there is a group of transitions whose names start with *null*. These transitions are those without corresponding actions and will fire automatically in the mapping rules.

```
▼Declarations
  ▼colset Agent = string;
  ▼colset maker = with maker;
  ▼colset Maker= product maker * Agent;
  ▼colset applicant = with applicant;
  ▼colset Applicant = product applicant * Agent;
  ▼colset customs = with customs;
  ▼colset Customs = product customs* Agent;
  ▼colset carrier = with carrier;
  ▼colset Carrier = product carrier*Agent;
  ▼colset computer_sys = with computer_sys;
  ▼colset Computer_sys = product computer_sys*Agent;
  ▼colset electronic_app = with electronic_app;
  ▼colset Electronic_app = product electronic_app*Agent;
  ▼colset Computer_Electronic = union C: Computer_sys + E:Electronic_app;
  ▼colset timeline = with timeline;
  ▼colset rea = union MC: Maker + AC: Applicant + CC: Customs + RC: Carrier;
  ▼colset REA = list rea;
```

Figure 9.4: *Type declarations in the CPN model of* RNN_{ex1}

(2) *Constitutive norms*
Suppose we have three business organizations *Company_A*, *Company_B*, *Company_C* respectively enacting the roles *Maker*, *Applicant*, *Carrier*, and one governmental organization *Dutch_customs* enacting the role *Customs*. In this case study, we focus on the regulative aspect of normative structure and make a simplification on the constitutive norms that the actions observed in the physical world are aligned with the actions specified in the organization. Therefore, the only link we need is the agent role relations. Based on the agent and role relations and RNN_{ex1}, we then obtain a norm net named NN_{ex1}.

9.1.3 Compliance Checking

To demonstrate the working mechanism of compliance checking, we implemented an example in CPN tools, as shown in Figure 9.5. The initial marking of the CPN model is obtained from the observation of the real system. Suppose there is an interaction plan in which *Company_B* decides to introduce some goods to the EU territory and then *Company_A* makes an electronic ENS and also authenticates the ENS. Meanwhile, *company_C* has containerized the goods. Thereafter, the *company_B* lodges the ENS to the *Dutch_customs*. 10 hours after the ENS has been lodged, the goods are loaded by *Company_C*. Correspondingly, a sequence of enabled steps is identified in the CPN model, denoted as the arcs between the set of nodes from 1 to 9 in Figure 9.5. These steps present the state transitions of the CPN model subject to the sequence of events described in the business process. For example, the event (*"company_B"*, *lodge_ENS*) will enable a step consisting of four binding elements. To check the compliance of the interaction plan in NN_{ex1}, we track the markings of the violated place v and the record place *vr8_9* in Figure 9.3 after each event occurs.

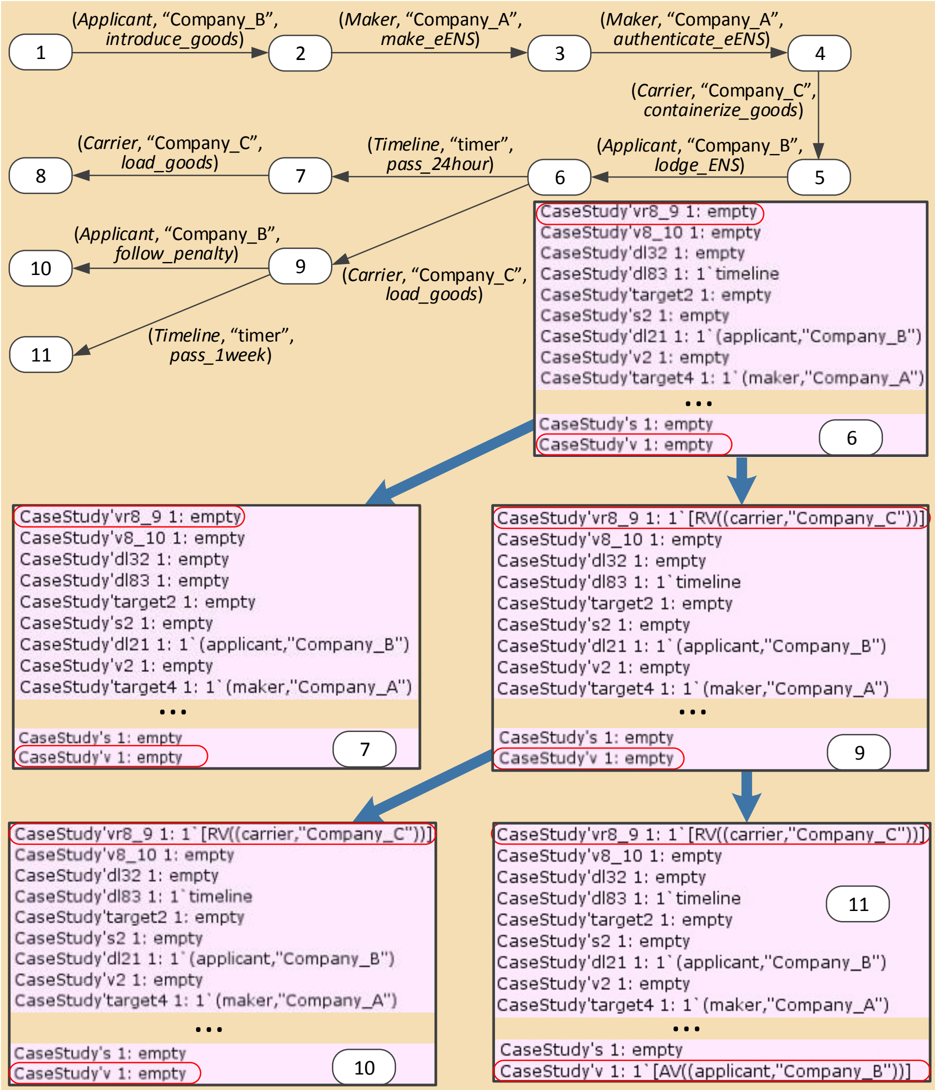


Figure 9.5: *Compliance checking in* NN_{ex1}.

It can be seen from the marking of node 9 (partially listed) that a violation is detected in the place *vr8_9* which relates to the *Carrier Company_C*. However, the place v is empty, indicating that NN_{ex1} is not in the violated state. The reason is that the violation can still be repaired by fulfilling the sanction in the

OE relation. From node 9, two further paths are formed by different events. The first path goes to node 10 by an event (*Applicant, "Company_B", follow_penalty*), while the second path goes to node 11 by an event (*timeline, pass_1week*). We can see that the place *v* remains empty in node 10 while containing a violation in node 11. That is, the first path repairs the violated norm by following the penalties before the deadline, while in the second path, the violation remains because the sanction is not fulfilled. In both cases, the system records the original violation of the *OE*-related norms. Obviously, the paths from node 1 to node 10 and node 11 are not full compliant since neither path stays empty in the place *vr8_9* and *v*. While if the event (*Carrier, "Company_C", load_goods*) would occur after the occurrence of the event (*timeline, pass_24hour*) at the moment of node 6, a full compliance would have been achieved following the path from node 1 to node 8.

9.1.4 Reflection

In this case study, we illustrated the process of modeling a set of legal regulations using the formalism of NNs, and the process of evaluating the compliance of agent interactions by means of the CPN models of NNs. The compliance evaluation results provide information on the compliance status of both individual actors' behavior and the collective behavior of the organization, which facilitates the identification of violation sources and possible ways of reparation. However, in the application of NNs, there are several issues that need to be considered.

- It is a large amount of work to interpret legal texts even with domain knowledge, especially when we are not only considering the contents of single regulations but also their interrelations.
- We use a specific role *Timeline* to represent a timer which can abstract the variation of time by actions such as *pass_1week*. In the mapping to CPNs, we don't use timed CPNs and hence cannot reason about temporal properties.
- It is not the case that the construction of every regulation is clear, i.e., its targets, conditions and deadlines. Usually, a single regulation represents a set of interrelated norms.
- Not all regulations have explicit conditions. Some of the regulations are unconditional and thus there is no representation of preconditions in the corresponding norms. Some of the regulations might have implicit conditions which relate to the components in other regulations. For this situation, we might need an overview of all the regulations to explore the condition and add an explicit representation in the norms.
- Not all regulations have explicit deadlines. Some of the regulations might have implicit deadlines relating to some other components in other regulations. For some other regulations, no deadline can be found, which indicates that the compliance or violation of the regulation can only be determined

at the end of the whole process. In this case, we set the deadline as the event representing the end of the interaction process.

9.2 Customs Declaration Management

9.2.1 Case Description

Customs declaration [61] is one of the crucial steps in international trade. Any organization whose business involves import and export activities needs a clear understanding of the imposed regulations, as they may affect the performance of the trading process in terms of lead time, cost and reliability. Typical activities involved in customs declarations are related to the registration, checking and control of movements of goods. Customs authorities and declarants are the two main parties involved in the customs declaration processes. The customs authorities have the responsibility of monitoring and controlling goods importation and exportation. Their main interest is to make sure that the import and export processes are in accordance with the legal regulations. On the other hand, the declarants have a strong demand of a fast and smooth customs declaration process since it can shorten the delivery time and improve the cost efficiency and reliability of their supply chains.

During the declaration process, changes may occur due to any differences between the declaration statement and the corresponding requirements. For example, the customs may find that the declaration is incorrect based on the inspection results, which may result in a correction of the declaration by the customs (and potentially will lead to a fine). To ensure customs compliance, a large number of regulations is imposed to restrict the behavior of both declarants and customs authorities. In this case study, we consider a summary of regulations reflecting a subset of the European Customs Community Code and the Implementing Provisions [60], in which there are about 350 articles. For some types of declarations more than 50 activities can be identified. To evaluate our approach, we use a subset of the customs regulations as a case study, described as follows.

R1. If the customs informs the declarant that the goods need to be physically controlled, the declarant must not make changes to the declaration.

R2. If the customs finds out any errors in the declaration, the declarant must not make changes to the declaration.

R3. After the customs releases the goods, changes to the declaration are not allowed.

R4. If customs debt is incurred and there is no temporary admission for the goods, the customs is forbidden to release the goods before the amount of the customs debt has been paid or the whole amount has been secured by the declarant.

R5. If the declarant submits an incomplete declaration, the declarant must include in the declaration the data for the identification of the goods and the amount of the goods, otherwise the customs should not accept the declaration.

R6. If the declarant submits an incomplete declaration and the customs releases the goods, the declarant should submit the supplementary documents within one month at most after the acceptance of the declaration.

This set of regulations set strict rules for both declarants and customs authorities. From the perspective of the customs authorities, it is their responsibility to ensure that the submitted declarations cannot be changed and the goods cannot be released before the customs debt has been paid or secured. From the perspective of the declarants, they have to make sure that the submitted declarations contain the required information and the supplementary documents are submitted on time in case of incomplete declarations.

In this scenario, customs always needs to react to different conditions. High ratio of mistakes and delays of customs operations may lead to financial losses for business companies and decrease the economic competitiveness of the country. To this end, customs authorities have a strong need for a declaration management system to handle all the processes from receiving the declaration, validating the declaration, performing risk assessment until clearing the declaration. As an example, Figure 9.6 shows a piece of business process for managing customs declaration modeling in BPMN [12]. This set of process is used by a customs agency (we refer to it by a name $Customs\_X$) to determine whether a received declaration complies with the set of validation rules which are applicable for accepting a declaration. The process starts when a declaration has been received and continues with a sub process of "validate declaration". Based on the validation results, the declaration will be processed differently.

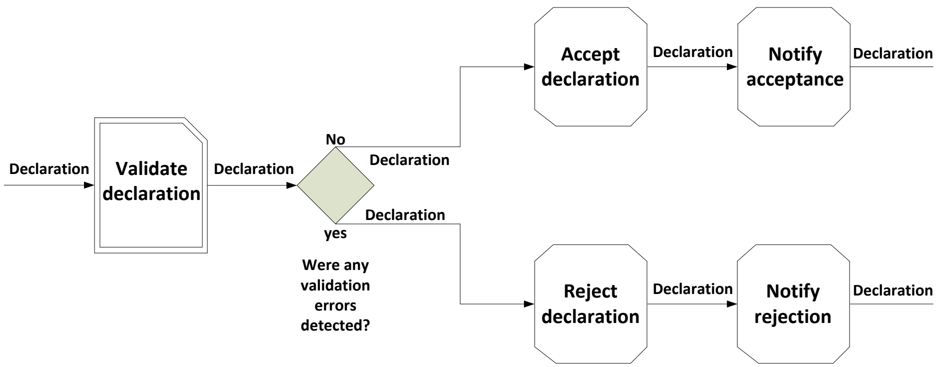


Figure 9.6: *A piece of business process for managing customs declaration.*

A crucial question of the customs authorities is whether the processes designed for the declaration management system are compliant with the legal regulations. Specifically, they have to verify such compliance before the processes can be implemented, otherwise undesired consequences might occur such as financial losses, unhealthy goods being released into free circulation, and environmental damages.

9.2.2 Formalization

(1) *Regulative norms*
In Section 9.2.1, we list six articles from the legislation. Each of these articles in these regulations can be represented by a regulative norm net. The process of modeling a single article with RNNs is basically the process of defining the deontic types, targets, deadlines, preconditions and logical connections between norms according to the semantics of the regulations. Currently the interpretation process is done manually and its correctness is highly dependent on the interpreter's knowledge of the legislation as well as the construction of RNNs. To this end, we consulted domain experts to confirm our interpretation which is formalized as a regulative norm net denoted by RNN_{ex2}.

For example, given the regulations *R1* and *R2*, we know that if the customs informs about the physical control on the goods (*inform_control*) the declarant is not allowed to change the declaration (*change_declaration*), and if the customs finds any errors in the declaration the declarant is not allowed to change the declaration as well. Accordingly, we can construct a norm net $RNN_1 = \text{AND}(n_1^r, n_2^r)$ where

- $n_1^r = (F, (Declarant,\ change\_declaration), (Customs,\ clear\_declaration), (Customs,\ inform\_control))$,
- $n_2^r = (F, (Declarant,\ change\_declaration), (Customs,\ clear\_declaration), (Customs,\ find\_errors))$.

According to **R5**, there is a special type of declaration submissions, through which the declarant is allowed to submit an incomplete declaration (*submit_incompDeclaration*). If this is the case, the declarant should include the identification information of the goods otherwise the customs should not accept the declaration, as specified by **R5** and represented by RNN_2=OE(n_3^r, n_4^r) where

- $n_3^r = (O, (Declarant,\ add\_identification), (Customs,\ accept\_declaration), (Declarant,\ submit\_incompDeclaration))$,
- n_4^r=(F, ($Customs$, $accept\_declaration$), ($Customs$, $clear\_declaration$), λ),

In a similar way, we formalize the rest of the regulations given in Section 9.2.1. As a result, we obtain the regulative norm net RNN_{ex2}. We then build the CPN model of RNN_{ex2}, reserving as a prime module, following the mapping rules illustrated in Section 5.2. The resulting CPN model is shown in Figures 9.7, 9.8 and 9.9.

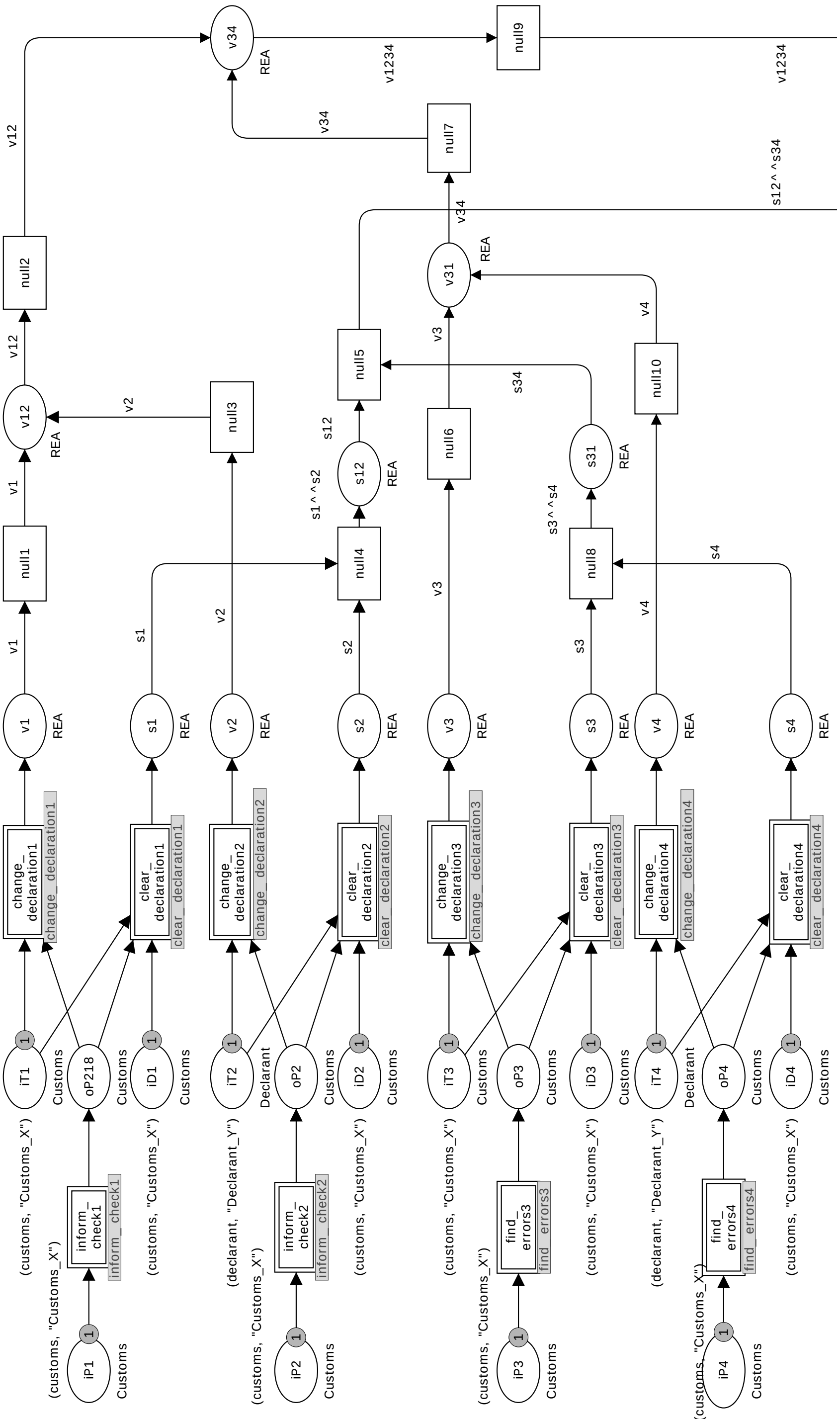


Figure 9.7: *The CPN model of RNN_{ex2} (continued in Figure 9.8).*

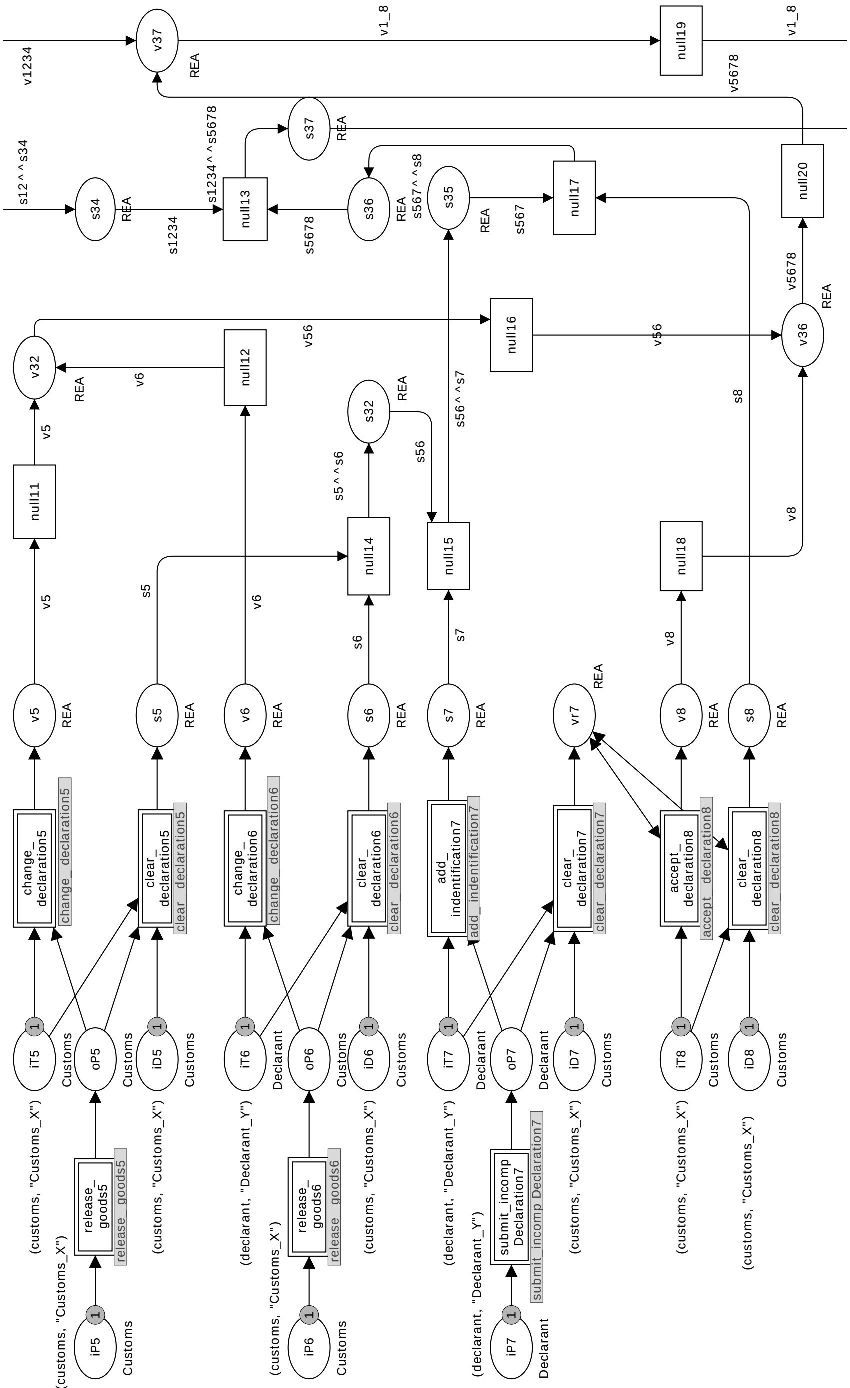


Figure 9.8: *The CPN model of RNN_{ex2} (continued in Figure 9.9).*

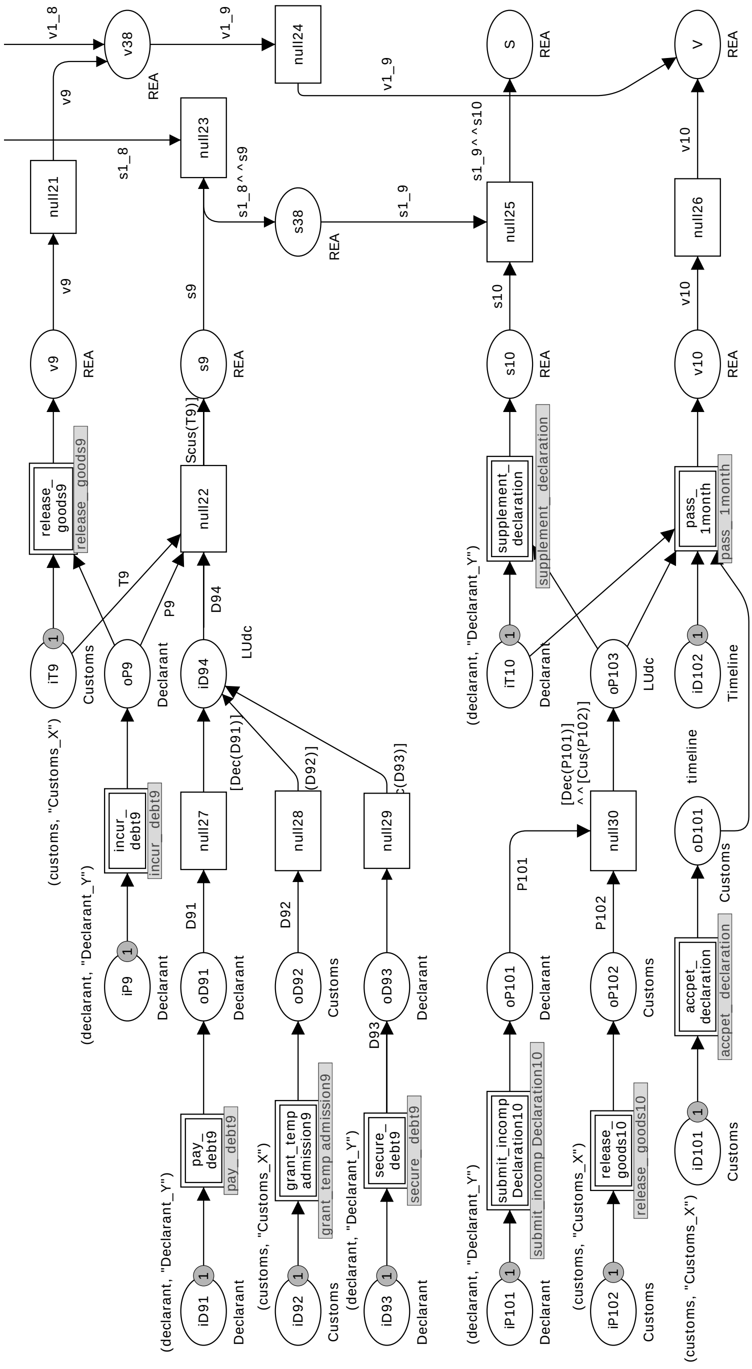


Figure 9.9: *The CPN model of RNN_{ex2} (continued from Figure 9.8).*

There are 10 norms in RNN_{ex2} and their CPN constructions can be distinguished by the labels of the places and arc expressions. Specifically, there are two sink places in this CPN model and they are respectively labeled S and V, indicating the overall complied and violated state of NN_{ex2}. With one OE relation in RNN_{ex2}, there is accordingly one place labeled $vr7$, indicating the repairable violated state. Moreover, there is a group of transitions whose label starts with *null*. These transitions are those without corresponding actions and will fire automatically in the mapping rules. The type declarations for the color sets used in the CPN model are shown in Figure 9.10.

```
▼Declarations
  ▼Types
    ▼colset Agent = string;
    ▼colset customs = with customs;
    ▼colset Customs = product customs * Agent;
    ▼colset declarant = with declarant;
    ▼colset Declarant = product declarant * Agent;
    ▼colset Timeline = with timeline;
    ▼colset Udc = union Dec: Declarant + Cus: Customs;
    ▼colset rea= union Scus: Customs + Sdec: Declarant;
    ▼colset REA = list rea;
    ▼colset LUdc = list Udc;
```

Figure 9.10: *Type declarations in the CPN model of* NN_{ex2}.

(2) *Constitutive norms*
To determine the normative consequence of the business process, we need to know the correspondences between the events specified in the business process and the events specified in the regulations, which corresponds to the constitutive part of a norm net. To do this, we first abstract all the events (role-actions pairs) specified in the regulations. Thereafter, we associate these events with their corresponding events in the business process. As a result, we obtain all the counts-as relations between the external events and the institutional events, which are captured by a constitutive norm net CNN_{ex2} containing 14 constitutive norms. Most of the constitutive norms are a straightforward mapping from the action of the agent *Customs_X* to the same action of the role *Customs*.

As an example, we show the CPN construction of n_3^c: the transition *find_error* in the CPN model of RNN_{ex2} is replaced by a substitution transition indicated in Figure 9.7 and assigned a submodule presented in Figure 9.11. Agents are represented by the tokens distributed in the CPN model, which are of two types. The first type of tokens are with the type of *Agent* while the second type of tokens are with the type of *Customs*. The former is indicated by the name of the agent "*Customs_X*" and the latter is indicated by a product of the name of the agent and the role it enacts (*Customs*, "*Customs_X*"). From the CPN model of this constitutive norm, we can see that as long as the agent *Customs_X* takes anyone of the three actions *report_validationErrors*, *report_majorDiscrepancy* and *report_minorDiscrepancy*, the substitution transition *find_error* will be fired.

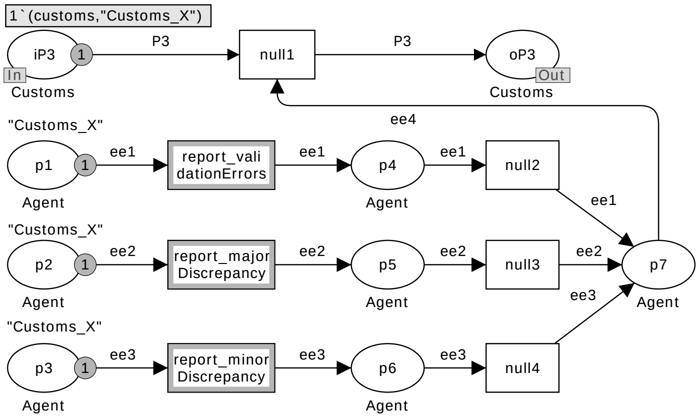


Figure 9.11: *The submodule of the substitution transition $find\_error$ in Figure 9.7.*

(3) *Interaction plans*
The whole declaration management process consists of a large number of decision points, activities and sub-processes. Our task in this step is to abstract all the events that are relevant to the control of the considered regulations and find their corresponding positions in the declaration management process. Once we know all the possible positions of the events in the process, we can organize them into interaction plans (sequences of events) according to the order of their appearances in the process. Later on we use these interaction plans as inputs to the CPN model of RNN_{ex2} to indicate the firing order of the transitions such that we can determine the compliance of the business process.

After discussing with the process designer, we abstract 7 interaction plans from a part of the declaration management process, which indicate different solutions for managing declarations according to different conditions. As an illustration, here we give one example that is of interest for the compliance checking results.

- IP_3 = ((*Customs_X*, *receive_declaration*), (*Customs_X*, *validate_declaration*), (*Customs_X*, *accept_decla-ration*), (*Customs_X*, *notify_acceptance*), (*Customs_X*, *asses_risk*), (*Customs_X*, *inform_check*), (*Customs_X*, *perform_check*), (*Customs_X*, *report_minorDiscrepancy*), (*Customs_X*, *accept_changes*), (*Customs_X*, *clear_-declaration*)).

Notice that in the interaction plans IP_3 there are 10 events while only 5 of these events have their counterparts in the regulations. The reason in this case is that we only take a part of all the relevant regulations. But in other cases, it could also be that not all the events specified in a business process are covered by the regulations, which means that they will not have any normative consequence.

9.2.3 Compliance Checking

Up to now, we have constructed the CPN model for the regulations and abstracted the interaction plans from the business process. The task then is to converge these two information flows such that we can verify whether the business process is complying with the regulations (consistency can be derived from the compliance results by checking whether any of the abstracted interaction plans is full compliant in this case study). To do this, we input each interaction plan into the CPN model such that the transitions corresponding to the events in the interaction plans will be fired in the order of their appearances. After the firing of each event, we can check whether there are tokens in the featured places of the CPN model to determine whether the business process is compliant with the regulations.

After running the checking procedure against all the interaction plans, we found two tokens in the output place labeled V in the CPN model of NN_{ex2}, as shown in Figure 9.12. This means that there is non compliances between the declaration management process and the regulations. The non compliances are caused by the event (*Declarant*, *change_declaration*) from the perspective of regulations. Going back to the interaction plans, we found an event (*Customs_X*, *accept_changes*) in IP_3 which is a corresponding event from the perspective of the business process. Accordingly, in the document of the business process, we found a description "The customs will hand over the control result to a specialist who will take a customs position by determining whether the control resulted in [...] a minor discrepancy, in which case he will ask an assistant to correct the declaration or instruct the declarant to do so."

Further investigation on the CPN model shows that this non-compliance is caused by the violation of norm n_2 relating to regulation $R2$. The reason is that the regulation says if the customs finds out any errors in the declaration, then the declarant is forbidden to change the declaration. However, in the business process, it is possible for the customs or the declarant to make changes to the declaration when there are only minor discrepancies. From the compliance checking results, we know that only one of the abstracted interaction plans is not compliant while the others are full compliant. Therefore, we can determine that the set of regulations is consistent.

9.2.4 Reflection

In this case study, we investigated the application of NNs for the problem of regulative compliance of business processes. By means of constitutive norms, the activities described in business processes are linked to the activities constrained by legal regulations. By means of regulative norms, the normative properties of

Figure 9.12: *Simulation results of interaction plan IP_3.*

the regulations are formalized. Based on the CPN mapping of NNs, we are able to computationally evaluate the regulatory compliance of business processes. From this case study, two major issues are reflected.Firstly, one needs to have a sufficient understanding of the design objectives of the business process when abstracting interaction plans from the business process. It is not the case that the process is just a simple set of activities. Secondly, regulations usually cover organizational behavior in a general way, while the business process is often designed from the perspective of a specific organization and thus might have a single view of the interactions with other organizations. This often leads to a need for concept alignments between the specification of the regulations and the specification of the business process.

With automated support for regulatory compliance, it is convincing that the workload of compliance departments can be reduced. Their work will change from multiple tasks related to compliance checking procedures to the focus on maintenance of regulation management system and analysis of the optimal process design. Moreover, organizations can make their business processes more reliable in terms of regulatory compliance, and in some cases can be used as evidences to show their compliance capabilities, e.g., the Authorized Economic Operator (AEO) program [59]. On the other hand, the control and maintenance cost of the governmental authorities can also be reduced, with the possibility of providing integrated services (e.g., regulation interpretation, consultation, and risk management etc.).

9.3 Food Importation

9.3.1 Case Description

The World Customs Organization (WCO) has defined a framework called the Authorized Economic Operator (AEO) program [59] in order to address the tensions created by the simultaneous growth in international trade and requirements for increased security. The European Communities' implementation of AEO permits various customs administrations to grant AEO certificates to qualified companies under which they enjoy special privileges. Taking the scenario of importing food from a country outside the EU to the Netherlands, a number of governmental authorities and companies are involved, which together form a virtual organization. Such a virtual organization is governed by different sets of regulations concerning different aspects of the food importation process. For example, the EU has a set of general regulations in which one is that the food authority is *obliged* to carry out a food quality inspection. With the introduction of the AEO programme, the Dutch government introduced new regulations for the specific domain of AEO-certified goods in order to improve trading efficiency. For example, one regulation is that a food authority is *forbidden* to carry out a food quality inspection, if customs has already done so. Additionally, companies such as container terminals play an important role and bring their own regulations, e.g., a regulation at one container terminal is that carriers are *obliged* to transport their goods thence within two days of unloading. Given these different sets of regulations, it is essential to capture not only their individual functionalities but also their interrelations concerning the governance of real world events.

9.3.2 Modeling Collective Institutions

In this case study, we consider three institutions catpured by three norm nets $\{\widehat{NN}_1, \widehat{NN}_2, \widehat{NN}_3\}$, whose governance scopes are based on three contextual dimensions {*Individuality*, *Location*, *Food*}. *Individuality* refers to four agents $\{ag_1$, ag_2, ag_3, $ag_4\}$. *Location* is provided with a set of values $\{t_a$, t_b, t_c, $w_a\}$ in which the first three elements represent three container terminals and the fourth element represents a warehouse. *Food* has a corresponding value set {AEO-beef, nAEO-beef} which respectively indicate importing beef with AEO certificate and without AEO certificate. Note that in this paper we only consider parts of the value set which are most relevant to our analysis.

The norm net model of the first institution $\mathcal{I}_1$ is described as follows.

- $\widehat{NN}_1 = (gs_1, RNN_1, CNN_1)$ where
  - $gs_1 = \{ag_1, ag_2, ag_3\} \times \{t_a, t_b, t_c, w_a\} \times \{AEO\_beef, nAEO\_beef\}$
  - $RNN_1 = AND(AND(n^r_{11}, n^r_{12}), n^r_{13})$ where
    - n^r_{11} = (O, (*Food_authority*, *inspect_quality*), (*Carrier*, *pass_border*), (*Carrier*, *arrive*))
    - n^r_{12} = (F, (*Carrier*, *unload_food*), (*Food_authority*, *inspect_quality*), λ)
    - n^r_{13} = (F, (*Carrier*, *choose_inspecL*), λ, λ))
  - $CNN_1 = \{n^c_{11}, n^c_{12}\}$ where
    - n^c_{11} = ((ag_1, *land_plane*)$\vee$(ag_1, *doc_ship*)$\vee$(ag_1, *stop_lorry*), (ag_1, *reach_border*)) $\rightarrow$ (*Carrier*, *arrive*)
    - n^c_{12} = ((ag_2, *land_plane*)$\vee$(ag_2, *doc_ship*)$\vee$(ag_2, *stop_lorry*), (ag_2, *reach_border*)) $\rightarrow$ (*Carrier*, *arrive*)
    - n^c_{13} = ((ag_1, *remove_goods*), (ag_1, *reach_border*)) $\rightarrow$ (*Carrier*, *unload_food*)
    - n^c_{14} = ((ag_2, *remove_goods*), (ag_2, *reach_border*)) $\rightarrow$ (*Carrier*, *unload_food*)
    - n^c_{15} = ((ag_1, *change_inspecL*), (ag_3, *inform_inspecL*)) $\rightarrow$ (*Carrier*, *choose_inspecL*)
    - n^c_{16} = ((ag_2, *change_inspecL*), (ag_3, *inform_inspecL*)) $\rightarrow$ (*Carrier*, *choose_inspecL*)
    - n^c_{17} = ((ag_1, *pass_border*), λ) $\rightarrow$ (*Carrier*, *pass_border*)
    - n^c_{18} = ((ag_2, *pass_border*), λ) $\rightarrow$ (*Carrier*, *pass_border*)
    - n^c_{19} = ((ag_3, *test_sample*)$\wedge$(ag_3, *asses_risk*), (ag_1, *reach_border*) $\vee$(ag_2, *reach_border*)) $\rightarrow$ (*Food_authority*, *inspect_quality*)

To operationalize the institution, we build a hierarchical CPN model for NN_1 following the mapping rules presented in 5.2. The prime modules of the hierarchical CPN models is shown in Figures 9.13. As for the submodules, we show as an example the models corresponding to the constitutive norms n^c_{11} and n^c_{19}, respectively shown in Figures 9.14 and 9.15.

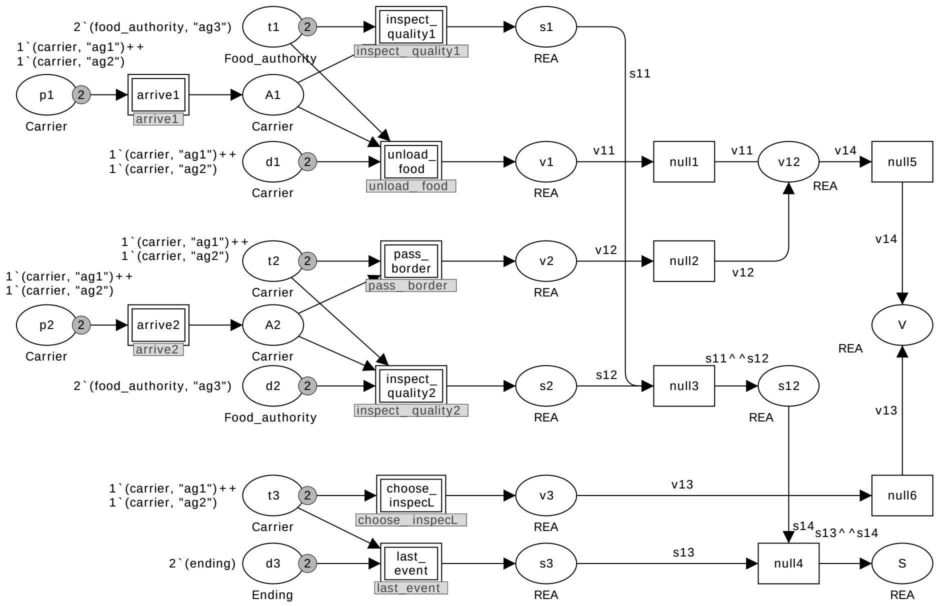


Figure 9.13: *The prime module of the CPN model of* $\widehat{NN}_1$.

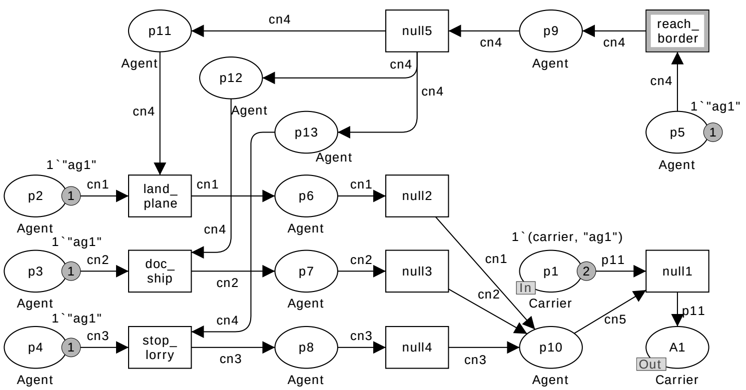


Figure 9.14: *The submodule of the substitution transition arrive*1 *in Figure 9.13.*

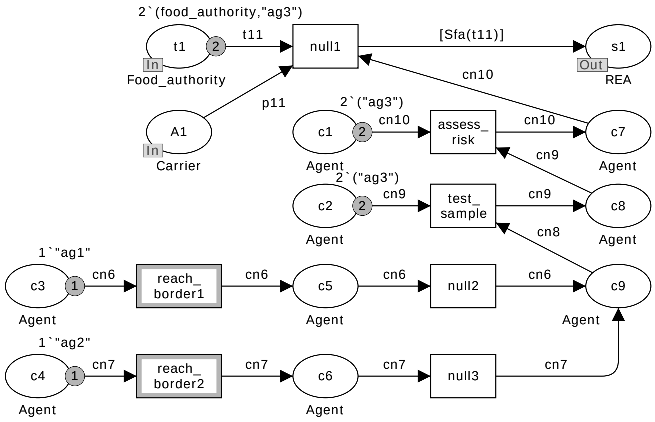


Figure 9.15: *The submodule of the substitution transition inspect_quality*1 *in Figure 9.13.*

Similarly, we give the norm net model of the second institution $\mathcal{I}_2$, and build a hierarchical CPN model. Figure 9.16 shows the prime module of the resulting hierarchical CPN model.

- $\widehat{NN}_2 = (gs_2, RNN_2, CNN_2)$ where
  - $gs_2 = \{ag_1, ag_2, ag_3, ag_4\} \times \{t_a, t_b, w_a\} \times \{AEO\_beef\}$
  - $RNN_2 = AND(n^r_{21}, n^r_{22})$ where
    - n^r_{21} = (F, (*Food_authority*, *inspect_quality*), λ, (*Customs*, *inspect_quality*))
    - n^r_{22} = (F, (*Carrier*, *unload_food*), (*Food_authority*, *inspect_quality*), λ)
  - $CNN_2 = \{n^c_{21}, n^c_{13}, n^c_{14}, n^c_{19}\}$ where
    - n^c_{21} = ((ag_4, *test_sample*)$\wedge$(ag_3, *asses_risk*), (ag_1, *reach_border*) $\vee$(ag_2, *reach_border*)) $\rightarrow$ (*Customs*, *inspect_quality*)

Finally, we give the norm net model of the third institution $\mathcal{I}_3$, and build a hierarchical CPN model. Figure 9.17 shows the prime module of the resulting hierarchical CPN model.

- $\widehat{NN}_3 = (gs_3, RNN_3, CNN_3)$ where
  - $gs_3 = \{ag_1, ag_2\} \times \{t_a\} \times \{AEO\_beef, nAEO\_beef\}$
  - $RNN_3 = OE(n^r_{31}, n^r_{32})$ where
    - n^r_{31} = (O, (*Carrier*, *leave_terminal*), (*Timeline*, *pass_2days*), (*Carrier*, *arrive*))
    - n^r_{32} = (O, (*Carrier*, *pay_fine*), (*Timeline*, *pass_1month*), λ)

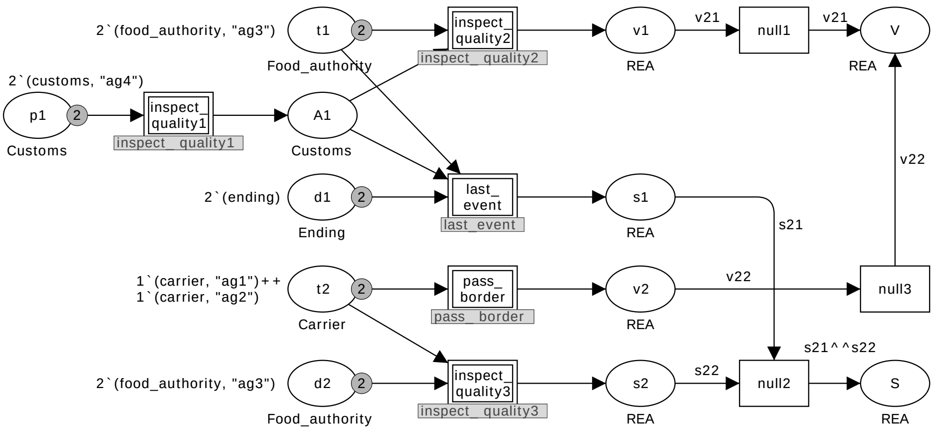


Figure 9.16: *The prime module of the CPN model of* $\widehat{NN}_2$.

– $CNN_3 = \{n^c_{31}, n^c_{32}, n^c_{33}, n^c_{34}, n^c_{13}, n^c_{14}\}$ where

  * $n^c_{31} = ((ag_1,\ pass\_border),\ \lambda) \rightarrow (Carrier,\ leave\_terminal)$
  * $n^c_{32} = ((ag_2,\ pass\_border),\ \lambda) \rightarrow (Carrier,\ leave\_terminal)$
  * $n^c_{33} = ((ag_1,\ pay\_cash) \vee (ag_1,\ pay\_credit),\ (ag_1,\ receive\_notice)) \rightarrow (Carrier,\ pay\_fine)$
  * $n^c_{34} = ((ag_2,\ pay\_cash) \vee (ag_2,\ pay\_credit),\ (ag_2,\ receive\_notice)) \rightarrow (Carrier,\ pay\_fine)$

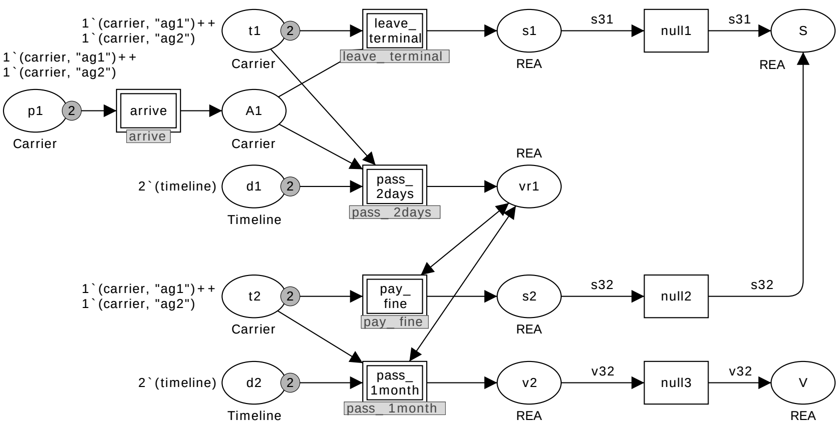


Figure 9.17: *The prime module of the CPN model of* $\widehat{NN}_3$.

The type declarations for the color sets used in the CPN models shown above are listed in Figure 9.18.

```
▼Declarations
  ▼colset Agent = string;
  ▼colset food_authority = with food_authority;
  ▼colset Food_authority = product food_authority * Agent;
  ▼colset customs = with customs;
  ▼colset Customs = product customs * Agent;
  ▼colset carrier = with carrier;
  ▼colset Carrier = product carrier * Agent;
  ▼colset Timeline = with timeline;
  ▼colset Ending = with ending;
  ▼colset rea = union Sfa:Food_authority+Sct:Customs +Scr:Carrier;
  ▼colset REA = list rea;
```

Figure 9.18: *Type declarations in the CPN model of* $\widehat{NN}_1$, $\widehat{NN}_2$ *and* $\widehat{NN}_3$.

In this case study, the three norm nets share three contextual dimensions *Individuality*, *Location* and *Food*, which are constrained by different sets of values. For example, $\widehat{NN}_1$ governs all European terminals and warehouses when importing food to the EU countries. $\widehat{NN}_2$ concerns the regulation of importing AEO certified food AEO_beef via Dutch terminals t_a, t_b and warehouse w_a, while $\widehat{NN}_3$ representing a terminal company regulates all food imports through terminal t_a only. Given the definition of the three norm nets and their CPN models presented above, we now illustrate how norm conflicts can be detected using the approach presented in Section 6.3 by means of the CPN tools.

9.3.3 Conflict Detection

Figure 9.19 shows a part of institutional evolutions with respect to three interaction plans (sequences of external events) in this case study. In general, when an external event occurs (shown above/below arrows), the first task is to identify which institutions have governance competence, and then identify which norms in these institutions are triggered. It can be seen that at different time instants, there are different sets of institutions that are activated concerning the occurrence of the event characterized by the attached contextual information. Each circle represents an institutional state, and a column of institution $\mathcal{I}_i$ above/below the circle indicates the collective institutions set at the time. For example, for the states 1 and 1′, the activation of $\mathcal{I}_3$ depends on whether the location where the carrier arrives is t_a. With more than one institutions being activated, different sets of regulative norms from different institutions are triggered to constrain the behavior, which might cause conflicts. For example, three norms from the three institutions are triggered simultaneously at 3, between which two conflicts occur. In this case study, there are in total four pairs of conflicts (indicated by a line with a cross) by providing four different interaction plans (event sequences).

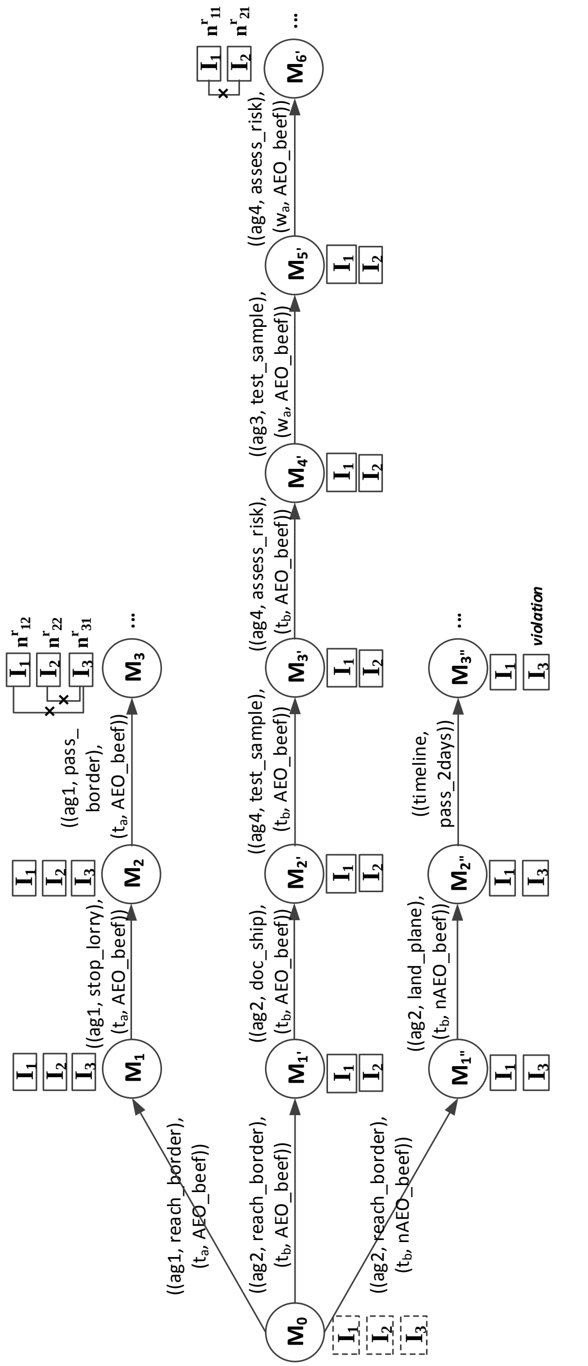


Figure 9.19: *Collective institution set and norm conflicts.*

As an example, we show how the normative states of the three institutions change along with the first interaction plan in terms of the markings of the corresponding CPN models, as shown in Figure 9.20. It can be seen that initially at the marking represented by node 1, all the overall satisfied and violated states (places labeled with S and V) of the three institutions are empty, highlighted by the red rectangles. From node 1, an institutional event (*Carrier*, *arrive*) is generated by the occurrence of the external events (*ag1*, *reach_border*) and (*ag1*, *stop_lorry*). As a result, another marking represented by node 2 is generated, in which the states of the three pairs of satisfied and violated places remain the same as that at node 1. From node 2, two institutional events (*Carrier*, *pass_border*) and (*Carrier*, *leave_terminal*) are generated by the occurrence of the external event (*ag1*, *pass_border*). The first institutional event is generated in both institutions $\mathcal{I}_1$ and $\mathcal{I}_2$, and the second institutional event is generated in institution $\mathcal{I}_3$. As a result, we obtain a new marking represented by node 3, in which the overall violated places of the first and second institutions both get a token (*carrier*, "*ag1*") while the overall satisfied places remain empty, indicating the agent $ag1$ enacting the role *carrier* produces a violation in both $\mathcal{I}_1$ and $\mathcal{I}_2$. However, the overall violated place of $\mathcal{I}_3$ remains empty but the overall satisfied place get a token (*carrier*, "*ag1*"), indicating $\mathcal{I}_3$ is satisfied with respect to the agent behavior. These imply that the three institutions give contradictory compliance evaluation results with respect to the same agent behavior (*ag1*, *pass_border*). Therefore, we can derive two norm conflicts, respectively between $\mathcal{I}_1$ and $\mathcal{I}_3$, and between $\mathcal{I}_2$ and $\mathcal{I}_3$.

9.3.4 Reflection

In this application case, we illustrated how collective institutions are modeled by norm nets to facilitate the detection of norm conflicts. The simulation results indicate that our conflict detection approach is able to identify conflicts among different institutions. Moreover, the specification of the governance scope of the institutions formalizes the control boundary of each institution and provides a better understanding of how the three institutions cooperatively work together in regulating agent behavior.

Since we only analyze a partial set of interaction plans (sequences of events), we cannot determine whether the conflicts found are weak conflicts or strong conflicts. As we discussed in Section 6.3, to check whether a conflict is weak or strong, we need to know all the meaningful traces of events that can occur in the corresponding real system, which is usually difficult. Nevertheless, if the aim is to check whether a business process will cause any conflicts with respect to an institution or a set of institutions, we can always abstract all the possible traces of events from the business process. In this way, we are able to make sure the compliance of the process design.

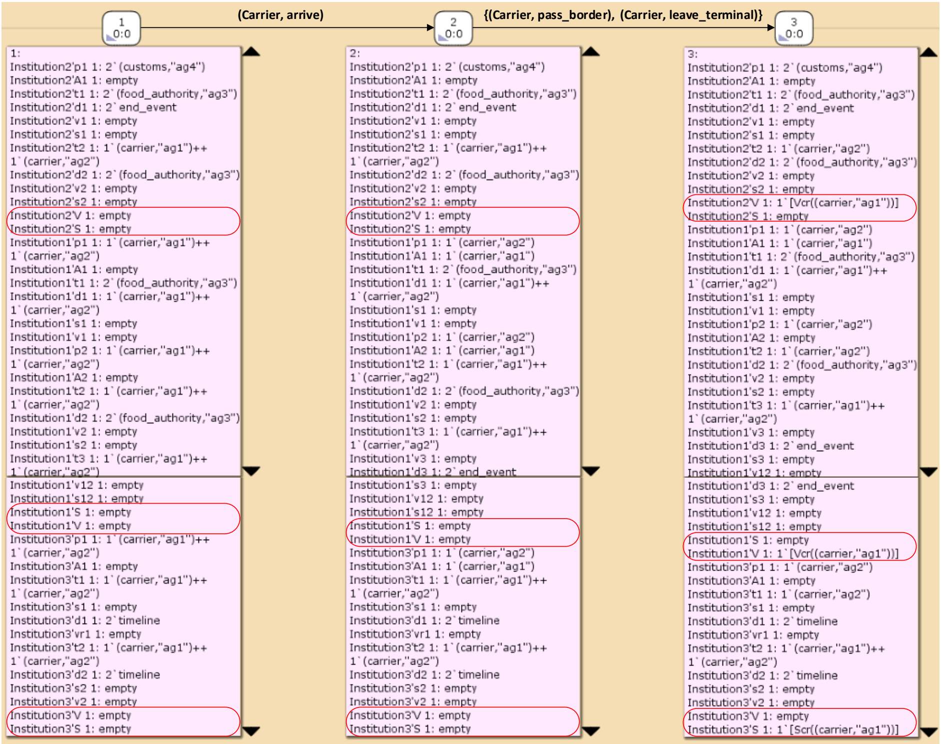


Figure 9.20: *Normative state transition of the three institutions.*

9.4 Conclusions

In this Chapter, we have shown several application cases related to the normative structure of OperA+. The first two case studies demonstrate from different aspects the formalization of compliance rules and the verification of agent behavior against the rules by means of Norm Nets (NNs) and the mapping to CPN models. In particular, we illustrated how CPN tools can be used to facilitate the compliance evaluation process. In the third application case, we investigated the formalism of contextualized NNs in the modeling of collective institutions. Furthermore, we illustrated the use of CPN mapping of NNs for the problem of norm conflict detection. The results of the case study show that our approach is able to capture the cooperative regulation in a multi-institution environment and identify conflicts among the norms imposed by different institutions.

Chapter 10

Application: Preference Structure

This chapter has the following contribution:

- an illustration of the construction of preference structures and the approach of combined-reasoning with norm compliance and preference satisfaction based on an example of a travel scenario.

10.1 Case Description

Assume that a group of three people *Alice*, *Bob* and *Carl*, together go for a business trip granted by their employer (organization X). For this trip, on the one hand, the three people have to follow a set of regulations that are prescribed to govern the behavior of the employees in organization X. On the other hand, each of the three people has his or her own preferences over the actions that can be taken. In order to evaluate a given set of interaction plans, both the regulations of the organization and the preferences of the individuals are important factors to be considered. That is, an interaction plan has to be evaluated with respect to whether the regulations are complied with and to what extent the individual preferences of the agents are satisfied.

10.2 Normative Constraints

The set of regulations are informally described as follows, in which we differentiated between a set of regulative norms and constitutive norms.

This chapter is based on our contribution to the 13th International Conference on Autonomous Agents and MultiAgent Systems, pp. 1373–1374, 2014 [122].

- *Regulative norms:*
  - *If the employer grants a travel to an employee, the employee should make a travel request before departure. Otherwise, the employee should not declare the travel cost.*
  - *If an employee makes a travel request, he or she should book the flight through the travel agency before departure, and should not book luxury hotels. Otherwise, the employee should not declare the travel cost.*
  - *After the application system sends a declaration notification, the employee should declare the cost before the application system informs the violation. Otherwise, after the application system informs about the extra cost, the employee has to pay the extra cost before the employer calculates the salary.*
- *Constitutive norms:*
  - *Organization X assigns a travel number given that Alice, Bob or Carl submit a application form, counts-as, an employer grants a travel.*
  - *Alice/Bob/Carl fills in a travel form and attaches a travel number, counts-as, an employee makes a travel request.*
  - *Alice/Bob/Carl departures, counts-as, an employee departures.*
  - *Alice/Bob/Carl submits a reimbursement form given that the Digi system emails a form after Alice/Bob/Carl departures, counts-as, an employee declares travel cost.*
  - *Alice/Bob/Carl sends an email to the Travel Partner (TP) and attaches the flight information, counts-as, an employee books a flight through the travel agency.*
  - *Alice/Bob/Carl books a 5⋆ hotel, counts-as, an employee books a luxury hotel.*
  - *The Digi system emails a form, counts-as, the application system sends a declaration notification.*
  - *The Digi system emails a violation, counts-as, the application system informs a violation.*
  - *The Digi system emails a bill, counts-as, the application system informs extra cost.*
  - *Alice/Bob/Carl pay by card after the Digi system emails a bill, counts-as, an employee pays extra cost.*
  - *Organization X calculates salary, counts-as, an employer calculates salary.*

Based on the description above, we construct a norm net $NN_{10} = (RNN_{10}, CNN_{10})$ where

- RNN_{10} = AND(OE(AND(AND(n^r_1, n^r_2), n^r_3), n^r_4), OE(n^r_5, n^r_6)) where
  - n^r_1 = (O, (*Employee*, *request_travel*), (*Employee*, *departure*), (*Employer*, *grant_travel*)),
  - n^r_2 = (O, (*Employee*, *book_agency*), (*Employee*, *departure*), (*Employee*, *request_travel*)),
  - n^r_3 = (F, (*Employee*, *book_luxuryHT*), λ, (*Employee*, *request_travel*)),
  - n^r_4 = (F, (*Employee*, *declare_cost*), λ, λ),
  - n^r_5 = (O, (*Employee*, *declare_cost*), (*App_system*, *inform_violation*)), (*App_system*, *send_notification*)),
  - n^r_6 = (O, (*Employee*, *pay_extra*), (*Employer*, *calculate_salary*)), (*App_system*, *inform_extra*)),
- CNN_{10} = {n^c_1, n^c_2, n^c_3, n^c_4, n^c_5, n^c_6, n^c_7, n^c_8, n^c_9, n^c_{10}, n^c_{11}, n^c_{12}, n^c_{13}, n^c_{14}, n^c_{15}, n^c_{16}, n^c_{17}, n^c_{18}, n^c_{19}, n^c_{20}, n^c_{21}} where
  - n^c_1 = ((*X*, *grant_travel*), λ) $\rightarrow$ (*Employer*, *grant_travel*),
  - n^c_2 = ((*Alice*, *fill_travelForm*) $\wedge$ (*Alice*, *attach_travelNo*), λ) $\rightarrow$ (*Employee*, *request_travel*),
  - n^c_3 = ((*Bob*, *fill_travelForm*) $\wedge$ (*Bob*, *attach_travelNo*), λ) $\rightarrow$ (*Employee*, *request_travel*),
  - n^c_4 = ((*Carl*, *fill_travelForm*) $\wedge$ (*Carl*, *attach_travelNo*), λ) $\rightarrow$ (*Employee*, *request_travel*),
  - n^c_5 = ((*Alice*, *departure*), λ) $\rightarrow$ (*Employee*, *departure*),
  - n^c_6 = ((*Bob*, *departure*), λ) $\rightarrow$ (*Employee*, *departure*),
  - n^c_7 = ((*Carl*, *departure*), λ) $\rightarrow$ (*Employee*, *departure*),
  - n^c_8 = ((*Alice*, *submit_reimburseForm*), (*Alice*, *departure*) $<$ (*Digi*, *email_form*)) $\rightarrow$ (*Employee*, *declare_cost*),
  - n^c_9 = ((*Bob*, *submit_reimburseForm*), (*Bob*, *departure*) $<$ (*Digi*, *email_form*)) $\rightarrow$ (*Employee*, *declare_cost*),
  - n^c_{10} = ((*Carl*, *submit_reimburseForm*), (*Carl*, *departure*) $<$ (*Digi*, *email_form*)) $\rightarrow$ (*Employee*, *declare_cost*),
  - n^c_{11} = ((*Alice*, *email_PT*) $\wedge$ (*Alice*, *attach_flightInfo*), λ) $\rightarrow$ (*Employee*, *book_agency*),
  - n^c_{12} = ((*Bob*, *email_PT*) $\wedge$ (*Bob*, *attach_flightInfo*), λ) $\rightarrow$ (*Employee*, *book_agency*),
  - n^c_{13} = ((*Carl*, *email_PT*) $\wedge$ (*Carl*, *attach_flightInfo*), λ) $\rightarrow$ (*Employee*, *book_agency*),
  - n^c_{14} = ((*Alice*, *book_5*$\star$), λ) $\rightarrow$ (*Employee*, *book_luxuryHT*),

– $n^c_{15} = ((Bob,\ book\_5\star),\ \lambda) \rightarrow (Employee,\ book\_luxuryHT)$,
– $n^c_{16} = ((Carl,\ book\_5\star),\ \lambda) \rightarrow (Employee,\ book\_luxuryHT)$,
– $n^c_{17} = ((Digi,\ email\_form),\ \lambda) \rightarrow (App\_system,\ inform\_declaration)$,
– $n^c_{18} = ((Digi,\ email\_violation),\ \lambda) \rightarrow (App\_system,\ inform\_violation)$,
– $n^c_{19} = ((Digi,\ email\_bill),\ \lambda) \rightarrow (App\_system,\ inform\_extra)$,
– $n^c_{20} = ((Alice,\ pay\_card),\ (Digi,\ email\_bill)) \rightarrow (Employee,\ pay\_extra)$,
– $n^c_{21} = ((X,\ calculate\_salary),\ \lambda) \rightarrow (Employer,\ calculate\_salary)$,

According to the construction of NN_{10}, we can then build a hierarchical CPN model according to the mapping rules presented in Chapter 5. Figure 10.1 shows the prime module of the CPN model. As for the submodules, we take n^c_8 as an example and show the corresponding CPN model in Figure 10.2.

10.3 Individual Preferences

From an individual perspective, everyone in this traveling group has his or her own preferences over the actions that can be taken:

- ***Alice*** *prefers booking a flight herself over booking a flight through the travel agency and over booking a train;*
- ***Bob*** *prefers booking a flight through the travel agency over booking a train if both Alice and Carl book a flight through the travel agency, and* ***Bob*** *prefers booking a 3⋆ hotel over booking a 4⋆ hotel if Bob book a flight through the travel agency;*
- ***Carl*** *prefers booking a 5⋆ over not booking a 3⋆ if Carl books a train, and* ***Carl*** *prefers booking a flight through the travel agency over booking a flight herself.*

Taking into account different levels of importance (indicated by numeric values), we exemplify the formalization of the preference specification of each person for this case as follows:

- **Alice**: $\{(Alice,\ (Alice,\ book\_self),\ true)(20) \gg (Alice,\ (Alice,\ book\_agency),\ true)(25) \gg (Alice,\ (Alice,\ book\_train),\ true)(30)\}$.
- **Bob**: $\{(Alice,\ book\_agency) \wedge (Carl,\ book\_agency){:}\ (Bob,\ (Bob,\ book\_agency),\ true)(30) \gg (Bob,\ (Bob,\ book\_train),\ true)(35),\ (Bob,\ book\_agency){:}\ (Bob,\ (Bob,\ book\_3\star),\ true)(18) \gg (Bob,\ (Bob,\ book\_4\star),\ true)(70)\}$
- **Carl**: $\{(Carl,\ book\_train){:}\ (Carl,\ (Carl,\ book\_5\star),\ true)(25)$
  $\gg (Carl,\ (Carl,\ book\_3\star),\ false)(70),\ (Carl,\ (Carl,\ book\_agency),\ true)(10)$
  $\gg (Carl,\ (Carl,\ book\_self),\ true)(50)\}$,

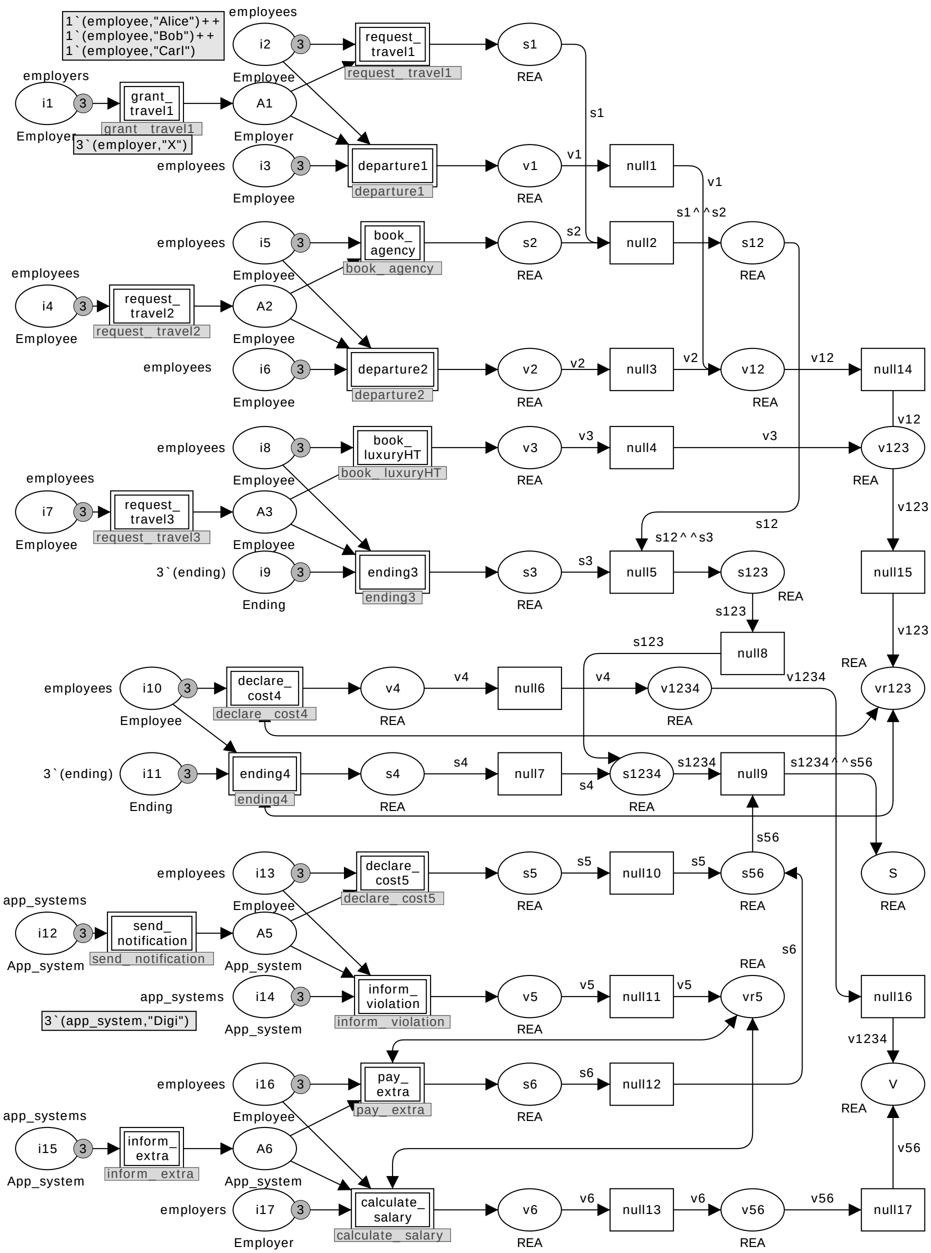


Figure 10.1: *Prime module of* NN_{10}

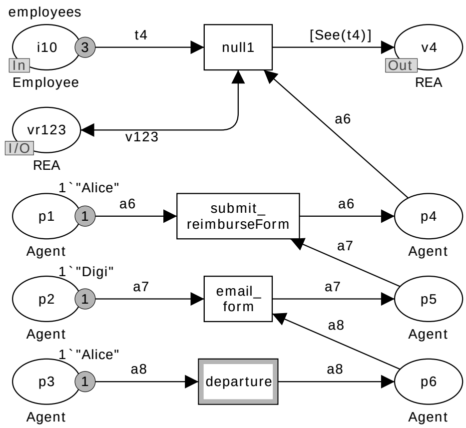


Figure 10.2: *Submodule of the substitution transition declare_cost4 in Figure 10.1*

Accordingly, we can build three CPN models for the preferences of the three people following the mapping rules presented in Chapter 7. The results are shown respectively in Figure 10.3, 10.4 and 10.5.

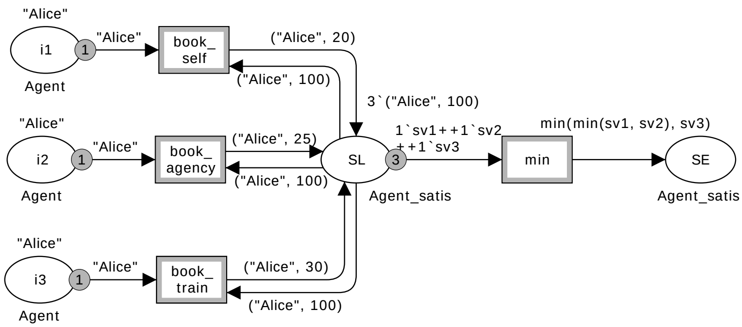


Figure 10.3: *CPN model representing the preferences of Alice*

The type declarations for the color sets used in the CPN models of the norm nets and preferences shown above are listed in Figure 10.6.

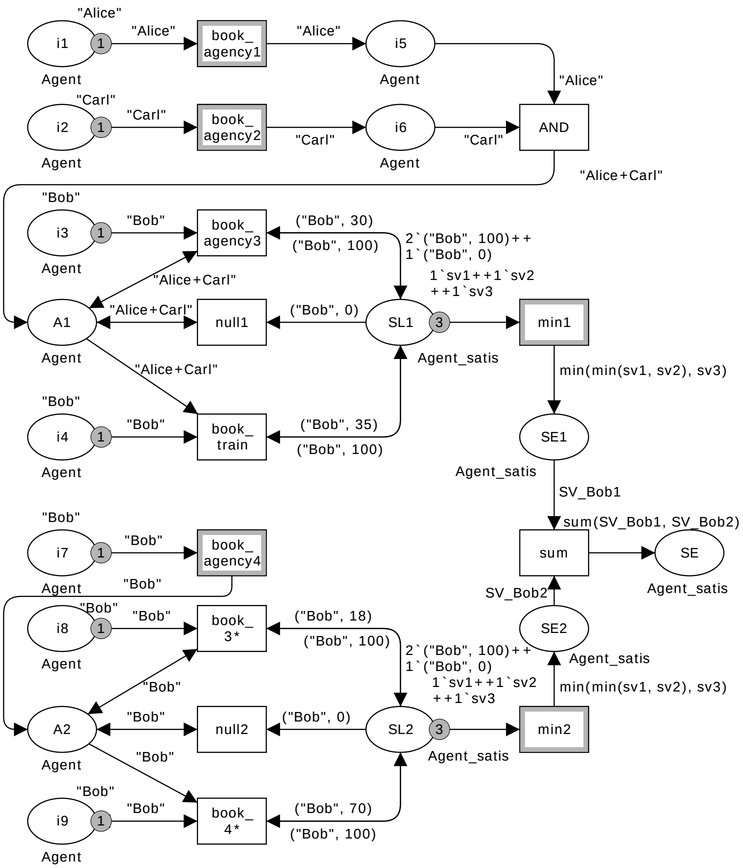


Figure 10.4: *CPN model representing the preferences of Bob*

```
▼Declarations
  ▼colset Agent = string;
  ▼colset Ending = with ending;
  ▼colset employer = with employer;
  ▼colset Employer = product employer * Agent;
  ▼colset employee = with employee;
  ▼colset Employee = product employee * Agent;
  ▼colset app_system = with app_system;
  ▼colset App_system = product app_system * Agent;
  ▼colset rea = union Ser:Employer +See:Employee;
  ▼colset REA = list rea;
  ▼colset Satis = int;
  ▼colset Agent_satis = product Agent * Satis;
```

Figure 10.6: *Type declarations in the CPN models shown in Figures 10.1 to 10.5.*

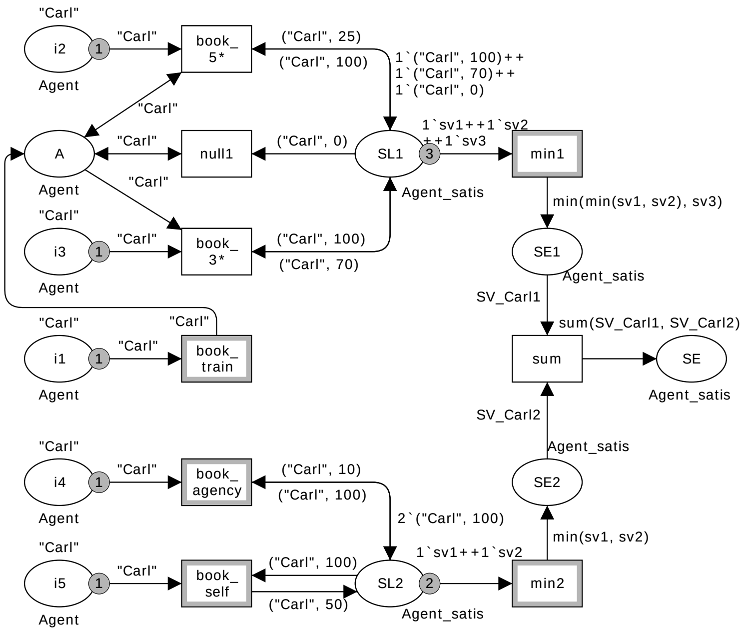


Figure 10.5: *CPN model representing the preferences of Carl*

10.4 Interaction Plans

Now suppose there are three possible interaction plans for the traveling group, as shown in Figure 10.7. Each plan starts with an action *grant_travel* corresponding to organization X, indicating that the employer grants the travel. Thereafter, each plan continues with four sequences of events with respect to the four agents, i.e., *Alice*, *Bob*, *Carl* and *Digi*. The ordering of the events is indicated by their positions relative to the time line at the bottom. Notice that the order of an event does not imply the real time of its occurrence. Moreover, we mix the actions of the events in the physical world and institutional events in the representation of interaction plans only to simply the description.

10.5 Combined Reasoning

With both the normative constraints and individual preferences defined, a combined reasoning of the three interaction plans can be obtained. To do this, we make use of the CPN models presented in Section 10.2 and 10.3, and follow the compliance checking approach presented in Section 5.4 and the preference evaluation approach presented in Section 7.4.3 to obtain the compliance status of each

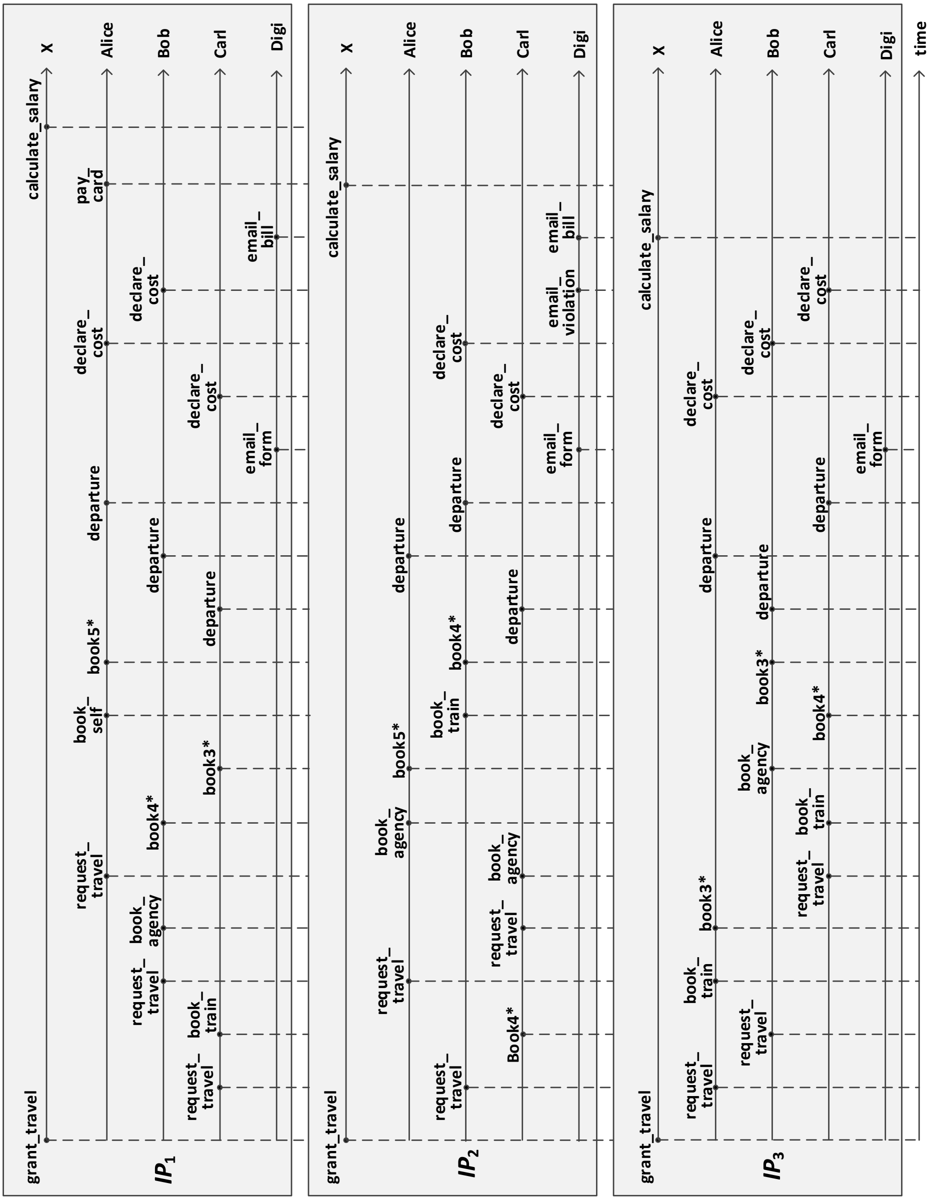


Figure 10.7: *Interaction Plans.*

plan and the satisfaction status of all the individuals. Finally, we get the results as shown in Table 10.1.

interaction plans	norm compliance	personal preferences		
		Alice	Bob	Carl
IP_1	C*	20	70	200
IP_2	V	25	35	10
IP_3	C	30	18	170

Table 10.1: *Compliance status and preference satisfaction of the three interaction plans*

The first plan is in sub-ideal compliance because *Alice* books the flight herself and a 5⋆ hotel but she pays the extra cost. The second plan is in non compliance since neither does *Alice* declare the cost nor pay the extra cost on time. According to the normative consequence, the third plan is mostly favorable since it is fully compliant with the norms. From the aspect of the satisfaction level, *Alice* prefers the first plan, *Bob* prefers the third plan, and *Carl* prefers the second plan. Notice that *Carl* has a satisfaction value *200* for the first plan because none of the alternatives in his two preferences are satisfied.

For individual agents, whether a plan is indeed the choice of an agent depends on the combined-reasoning strategy of the agent. For example, if *Alice* is *selfish*, her choice would be the first plan since she will try to maximize her preference satisfaction without considering the normative consequence. If *Alice* is *norm-aware*, the third plan would be her choice since she will try to minimize the violations.

From an organizational perspective, on top of norm compliance (both full and sub-ideal compliance), if the agents are situated in a horizontal organization, the third plan would be a favored choice for the whole group. Assuming that *Alice* is the leader of the group, if the agents are situated in a hierarchical organization, the first plan would be the choice.

10.6 Conclusions

In this chapter, we demonstrated the application of the preference structure of OperA+ and the approach of combined reasoning with norms and preferences, through an example of a travel scenario. We used a simple example to illustrate the working mechanism of the approach because the basic techniques are similar to that of the compliance evaluation in NNs we have shown before. The approach can be applied to more complex domains such as supply chains and subcontracting, where a number of actors collaborate with each other under sets of regulations. With possibly different values and interests, these actors might have varying preferences over the actions they can take.

At an agent level, we have shown how agents may be assisted in selecting alternative plans considering both the normative constraints and personal preferences with respect to their combined-reasoning strategies. At an organizational level, we have shown how combined-reasoning strategies may be adopted to evaluate alternative plans from a global perspective. In this application case, we assumed the preferences of all the individuals in the traveling group, which enables an analysis of the agents' interactions from an organizational perspective. In practice, however, it may be difficult to acquire the preferences of individual agents, especially when privacy issues are concerned. Therefore, the organizational level analysis of preference satisfaction, which heavily depends on the availability of individual agents' preferences, need to seek for means of reasoning with imperfect information.

Chapter 11

Conclusions

This dissertation focused on the design of organizational models that support dynamic and autonomous interactions in regulated environments. The motivation for such models came from the need to devise and evaluate solutions for ensuring organizational compliance that consider both the stability of organizational control and the flexibility of individual operation.

The OperA+ framework proposed in this research, uses the paradigm of normative multi-agent systems as the basis of conceptual design. At an abstract level, the concept of agents can refer to any autonomous entity including humans, software and organizations, and the concept of norms can refer to any explicit rules that govern the behavior of those autonomous entities. However, throughout the dissertation, we have used theoretical models of agents, agent interaction and norms to build models for organizations. The focus of OperA+ is not only the design of the regulated environment where the individual entities that form the organization interact, but also the representation of the preferences of those individual entities, such that a balance can be obtained between the stability of organizational control and the flexibility of individual operation.

This final chapter discusses the results of this research in Section 11.1, and identify directions for future work in Section 11.2

11.1 Discussion of Results

There are three main assumptions that underlie this research. Firstly, the development of an organization has the aim of accomplishing some global objectives that can be realized by coordinating the behavior of some individual participants. Secondly, the individual participants are autonomous entities with personal interests and needs, without necessarily sharing the same objectives of the organization in which they play a role. Finally, in order to ensure the achievement of the global objectives, the establishment of organizational compliance is needed to guide and regulate the behavior of the participants and their interactions. These assumptions are the basis to the research questions formulated in Chapter 1, which are listed as follows:

1. What are the constructs and structures that are needed for the representation of organizational compliance in organizational design?
2. How can we evaluate the actual behavior of the organizational participants with respect to the compliance requirements of organizational regulation?
3. How can we evaluate the consistency of different sources of compliance requirements that are relevant to the regulation of organizational behavior?
4. How to take into account the autonomy of individual participants in the evaluation of organizational compliance?

In the remainder of this section, we look back at the realized work, and discuss the main findings related to these questions.

11.1.1 Representing Organizational Compliance

Our first research question relates to the identification of constructs and structures that are needed for representing organizational compliance. In Chapter 1, we have identified the following main requirements that, in our opinion, should guide the establishment of organizational compliance for (virtual) organizations:

- coverage of multiple levels of the participants' activities and interactions through the roles they enact in the organization (compositionality)
- reflection of the important features borrowed from or impressed upon by the context to which the organization must adapt (contextualization)

The OperA+ framework presented in Chapter 3 was designed to fulfill both these requirements. By providing a multi-level and multi-context structure of organizational design, we demonstrate how compositionality and contextualization are embedded in the representation of organizational compliance. The multi-level approach enables the description of increasingly more detailed specifications for organizational compliance. The multi-context approach enables the explicit representation of the possible contexts to which the achievement of organizational compliance must adapt. The combination of these two approaches provides a contextual link between organizational values and implementation details. This not only enables a gradual understanding of how the dynamics of organizational compliance is coupled with the changes of applying contexts, but also facilitates the reuse of control components for ensuring organizational compliance across different contexts.

At a single level and within a single context, three structures are identified to be needed for the representation of organizational compliance. Firstly, the *social structure* presented in Chapter 4 provides a way of encapsulating compliance requirements of organizations through the specification of roles in the organization and role enactments of the participants. In OperA+, a role can be enacted by an organizational agent which can further elaborate the role into an organization of more finer-grained components (roles). Such a design choice enables specification of open systems in which the participants can make further decisions on how an accepted role is divided and assigned. In this way, organizational design can be performed in a distributed manner, i.e., organizations (participants in a larger organization) can take the responsibility to refine different parts of organizational design at a high abstraction level into more concrete ones. Secondly, the *normative structure* presented in Chapter 5 and 6 describes the compliance rules that govern the behavior of the participants in terms of the roles they enact in the organization. Finally, the *preference structure* presented in Chapter 7 describes the preferences of the individual participants. The social structure and normative structure indicate the requirements of organizational control while the preference structure indicates the the autonomy of individual participants, in achieving organizational compliance. Thus, the three structures together provide the constructs through which we are able to find the right balance between the level of control of the organization and the level of autonomy of the participants.

11.1.2 Evaluating Individual Compliance

The normative structure of an OperA+ model presents, from an organizational perspective, the expected behavior of the organizational participants in achieving organizational objectives. Based on the notion of norms (obligations, prohibitions, and permissions), we proposed a normative language Norm Nets (NNs) in Chapter 5 for the construction of normative structures in OperA+. NNs are not only capable of modeling single norms, but also provide the constructs to capture the interrelations between norms in a modular way. Moreover, by differentiating between constitutive norms and regulative norms, NNs are able to create a link between the occurrences of participants' behavior in the physical world and the perceived behavior of organizations. In a nutshell, a NN presents

- a set of regulative norms (conditional obligations/prohibitions with deadlines) and their interrelations (conjunction, disjunction, reparation/sanction) to regulate the behavior of the organizational participants in terms of the roles they take in the organization, and
- a set of constitutive norms (counts-as statements) to translate the behavior of the participants in the physical world into the behavior that is relevant to the control of the organization.

The specification of NNs gives a complete illustration of how participants' behaviors are interpreted by the organization and how such behaviors collectively impact the compliance status of the organization.

NNs specify at a conceptual level the expected behavior of the organizational participants for the achievement of organizational compliance. While the participants are autonomous entities who are capable of controlling their own actions, it is necessary to check whether the actual behavior of the participants is in accordance with the expected behavior desired by the organization. To this end, a mapping from NNs to Colored Petri Nets (CPNs) is presented in Chapter 5, which captures the structural as well as the behavioral properties of NNs. As a result, we are able to computationally check whether the behavior of the participants is compliant with the normative specification. Moreover, we are able to make use of the state space analysis techniques of CPNs for the analysis of compliance properties of organizational behavior.

Together with its mapping to CPNs, NNs provide a way of formalizing compliance rules and their interrelations, and operationalizing the evaluation of participants' actual behavior with respect to the obligations and prohibitions of the compliance rules. The specification of the interrelations between norms in NNs enables us to do more sophisticated evaluation of the participants' behavior, e.g., with respect to ideal and sub-ideal compliance. The evaluation results present evidence about whether organizational compliance is actually achieved. In case of non-compliance, the participants who have misbehaved can be identified by checking the states (or markings) of CPN models. Moreover, by using CPNs, we also benefit from the tool support in the CPN community, which facilitates the evaluation process.

11.1.3 Evaluating Regulation Consistency

Besides the evaluation of the compliance of individual participants, another aspect of compliance we have investigated relates to the consistency between the compliance rules coming from different regulation sources, e.g., internal policies, legal regulations, business agreements. This is an important issue especially in the setting of virtual organizations where multiple institutions are employed for the regulation of the participants' behavior.

Based on the observation of human society, we regard the process of designing an institution as not only the definition of a set of norms but also the characterization of its governance scope, i.e., what kinds of situations are under control of the institution, since this is what gives the institution its 'footprint'. To this end, in Chapter 6 we presented an extension of the normative model NNs described in Chapter 5 such that the governance scope of an institution can be formally captured. On the basis of the extended normative model, we further illustrated how multiple institutions cooperatively work together to regulate the behavior of the actors in a virtual organization, which then leads us to the problem of norm conflicts. Such a setting of multiple institutions distinguishes our approach of dealing with norm conflicts from the others, since the existence of normative conflicts is then not just by the definition of mutually exclusive deontic expressions but also the overlap of governance scopes. Furthermore, we presented a formal analysis of the definition of norm conflicts using the model of NNs. To detect the conflicts, a computational mechanism based on the CPN models of NNs is provided. In this way, we are able to check the consistency of the norms within an institution as well as the norms across different institutions.

In law systems, there are typically several layers of legislation and rule-giving from the constitutional one to the most concrete one of judicial interpretations. Such a characteristic is realized in the design of the normative structure of OperA+ models through contextualization of norms from abstract to concrete. In Chapter 6, we illustrated how a set of abstract norms captured in NNs can be refined with respect to the refinement of the contexts where the norms are applied. The link between the concepts used in abstract norm nets and the ones used in contextual norm nets is realized through counts-as statements. In this way, the construction of normative structures is extended to multiple levels of abstraction and the construction process gives a reflection of how changes of context lead to the structuring of the norms that govern the behavior in the context.

11.1.4 Integrating Individual Autonomy

Organizational participants are the actors upon whom organizations depend to achieve organizational objectives and ensure organizational compliance. Compliance refers to the regularized aspects of organizational design which provide a description of the legitimate ways that participants should follow in their collective pursuit of organizational objectives. While participants are autonomous entities with personal preferences, they are capable of controlling their own actions to pursue personal satisfaction and involvement. In such cases, it is important to take into account individual preferences in the design and evaluation of solutions to achieve organizational compliance.

In Chapter 7, we presented a preference language to capture various types of preferences of individual actors (participants), through which the preference structure of OperA+ models can be built. The preference language is built over the actions that the actors may take. It not only allows preference specifications over an actor's own actions but also over the other actors' actions and the actions relating to the roles specified in the organization. Different degrees of preference are indicated by numeric values that are attached to alternative courses of actions. Moreover, we investigated how actors may adapt their abstract preferences to the concrete contexts they encounter. The specification of the preferences of an actor implies the possible decision of the actor when there are alternative choices of actions. The decision making of the actor largely depends on the extent to which its preferences are satisfied. In order to operationalize the evaluation of preference satisfaction, we proposed a mapping of the preference language to CPNs such that we can computationally obtain to what extent a preference is satisfied with respect to the actions of the actors in the organization.

In OperA+, the preference structure presents the personal needs of individual participants while the normative structure presents the compliance requirements of organizational control. Both structures reflect important aspects that should be considered in the devise and evaluation of solutions for ensuring organizational compliance. To this end, we proposed a mechanism of integrating preference reasoning with normative evaluation, such that these two sources of information can be analyzed in a unified way. From an individual perspective, this can provide the necessary information for actors to make decisions considering both normative constraints and personal preferences. From an organizational perspective, if we have the knowledge of all the actors' preferences in the organization, the overall evaluation of the actors' interactions can be elaborated.

11.2 Future Work

Our ideas for future research concern the link between abstract and concrete concepts in the ontology of organizational models, practical issues of compliance and preference, emergence of norms, and unification of concepts and incorporation of mechanism design, which are detailed in the remainder of this section.

11.2.1 Linking Abstract and Concrete Concepts

One of the extensions of OperA+ as compared to OperA is that it extends organizational modeling to multiple abstraction levels. Accordingly, the ontology of organizations is built with increasing details, from concepts relating to abstract values, to concepts relating to concrete implementation. It is important to provide an integrated framework for linking abstract and concrete concepts and facilitate the reasoning of organizational behavior at different abstraction levels. To this end, further exploration of CPNs is needed, which will lead to a potential approach for the construction of multi-level organizational models. One direction is to formalize the hierachical structure of roles by encoding them into color sets or data types in CPNs.

11.2.2 Putting Compliance and Preference into Practice

In Chapter 5 we presented a normative language NNs for the representation of compliance rules. The types of compliance rules that can be dealt with by NNs in this dissertation are limited to the control-flow related compliance rules presented in [174]. Given the variety of compliance rules in practice, it is important to have a rich formalism that is capable of representing different aspects of compliance rules, e.g., time, resource, etc.

The current CPN-based computational model for NNs can be improved in many directions. One direction is to apply methods like the one presented in [176] to label transitions with costs so that the degree of compliance can be elaborated for each event sequence. Based on the degree of compliance, better alternative processes can be recommended to stakeholders or decision makers. A further step is to devise approaches to associate the costs on transitions with richer information in event logs, as similar to the methods proposed by [137, 136].

As mentioned in Chapter 7, our preference language is an extension of the work presented in [210], in which agent preferences are specified only over the actions of agents. In practice, actions may relate to other types of information such as resource usage, and therefore agents might need to specify their preferences over the consumption of resources. For instance, one may have a preference of buying cheaper tickets over traveling in less time. To this end, further investigation is needed to incorporate more types of information in the specification of preferences. The combination of different preferences in the satisfaction evaluation is now accomplished by a simple function of summation, which may limit the use of the model. It is necessary to have a deeper research on the possible relations between preferences and employ more sophisticated combination strategies. For example, a possible relation between two preferences might be that the fulfillment of one preference will cancel the other preference. As for the combination strategy, weights may be assigned to different preferences to represent their priority in the choice of an agent.

Furthermore, one of the main challenges in norm compliance is to bridge the gap between informal descriptions and formal specifications [147, 184, 173]. Therefore, a direction for future research is to integrate approaches of (automated) norm extraction and elicitation with formal normative languages such as the one proposed in this dissertation. Similarly to norms, individual preferences are usually specified in natural languages or more often hidden in the traces of events occurred. In the literature, there has been a lot of research on mining and eliciting preferences of actors (e.g., [107], [125]). However, how to integrate those approaches with formal preference models still needs further research.

11.2.3 Emergence of Norms

In this dissertation, we assume that compliance rules can be obtained from various regulation sources such as legislation, guidelines, policies, etc. The main focus is on the representation of such rules and the evaluation of agent behavior against these rules, following a top-down manner of norm enforcement. From the perspectives of both organizations and individuals, it is important to learn from interaction experiences. For example, a new participant needs to learn about the social conventions in an organization by finding behaviors that are (not) favored among the other participants. An organization may need to adopt a new norm to enhance its performance by restricting certain behaviors. In these cases, how norms should be automatically generated and evaluated are important questions to explore. Such issues on norms are closely related to the studies in the field of norm emergence (e.g., [191, 190]) where game-theoretic frameworks are often employed. Learning from log data, process mining techniques [1] have been succesfully used for model discovery. Integration of such approaches would provide a bottom-up support for norm compliance.

11.2.4 Unifying Concepts and Introducing Mechanisms

There has been a large number of normative models presented in the literature, which are grounded in different theories. A challenge for the whole normative community is to devise a unified framework that can be used to facilitate the communication and comparison between these different models. For example, though there are a lot of similarities between NNs and the normative models based on commitment life cycles, it is still difficult to map from one to the other. Besides, given the link between the notion of norms in normative systems and compliance rules in process mining, it would be a promising task to transfer the research results between these two domains.

Another promising direction of future research is to regard norms as a mechanism for agents to coordinate themselves and incorporate game-theoretic techniques for the analysis of agent behavior [25]. For example, by using games, we can explore the types of properties that a normative system should satisfy in order to be effective as a mechanism to obtain desirable behavior. This will enable a new perspective on how organizational control and individual autonomy can be analyzed and evaluated in a unified way.

Appendix A

List of Algorithms

In this appendix we present the algorithms of generating CPN models of Norm Nets, norm compliance checking and norm conflict detection.

A.1 References of CPN Elements

In order to reference to some special elements in a CPN model $\mathcal{N}$, we introduce the following notations:

- $SCP_{\mathcal{N}}$: the set of source places in $\mathcal{N}$,
- $SKP_{\mathcal{N}}$: the set of sink places in $\mathcal{N}$,
- IT_p: the set of input transitions of a place p in $\mathcal{N}$,
- OT_p: the set of output transitions of a place p in $\mathcal{N}$,
- IP_t: the set of input places of a transition t in $\mathcal{N}$,
- OP_t: the set of output places of a transition t in $\mathcal{N}$,
- LB_p: the label of a place p in $\mathcal{N}$,
- LB_t: the label of a transition t in $\mathcal{N}$,

In the following algorithms, italic statements follow the CPN syntax.

A.2 Mapping from an event to CPNs

Algorithm 9 gives the procedure of generating the CPN model of a given event (external event or institutional event).

Algorithm 9 CPN mapping of an event: Event_CPN

Require: ϵ ▷ An external or institutional event
Ensure: $\mathcal{N}$ ▷ A CPN model
$(ra, \alpha) \leftarrow \epsilon$
 ▷ (1) Create a transition representing the action α in the event
$T_{\mathcal{N}} \leftarrow \{\alpha\}$
 ▷ (2) Create two places representing the state whether the event occurs
$P_{\mathcal{N}} \leftarrow \{p_i, p_o\}$
$A_{\mathcal{N}} \leftarrow \{(p_i, \alpha), (\alpha, p_o)\}$
 ▷ (3) Assign color sets to the places according to the type of the event ϵ
colset Agent = *string* ▷ create a color set representing agents
if isExternalEvent(ϵ) **then**
  $\Sigma_{\mathcal{N}} \leftarrow \{Agent\}$
  $C_{\mathcal{N}} \leftarrow \{(p_i, Agent), (p_o, Agent)\}$
end if
if isInstitutionalEvent(ϵ) **then**
  colset r = *with* $name_{ra}$ ▷ create a color set representing a role
  colset r_ag = *product r* $*$ *Agent* ▷ create a color set representing agents enacting a specific role
  $\Sigma_{\mathcal{N}} \leftarrow \{Agent, r, r\_ag\}$
  $C_{\mathcal{N}} \leftarrow \{(p_i, r\_ag), (p_o, r\_ag)\}$
end if

A.3 Mapping from an event formula to CPNs

Algorithm 10 gives the procedure of generating the CPN model of a given event formula which consists of a set of events (external and institutional) combined with the three operators $\wedge$ (and), $\vee$ (or), $<$ (before).

Algorithm 10 CPN mapping of an event formula: EventFormula_CPN

Require: ψ ▷ An event formula
Ensure: $\mathcal{N}$ ▷ A CPN model
▷ The combined relation is $\wedge$ (and)
function AND_CPN(ψ_l, ψ_r)
  $\mathcal{N}_1 \leftarrow$ EventFormula_CPN(ψ_l)
  $\mathcal{N}_2 \leftarrow$ EventFormula_CPN(ψ_r)
  $P_{\mathcal{N}} \leftarrow P_{\mathcal{N}_1} \cup P_{\mathcal{N}_2} \cup \{p_\wedge\}$
  $T_{\mathcal{N}} \leftarrow T_{\mathcal{N}_1} \cup T_{\mathcal{N}_2} \cup \{null_\wedge\}$
  $A_{\mathcal{N}} \leftarrow A_{\mathcal{N}_1} \cup A_{\mathcal{N}_2} \cup \{(SKP_{\mathcal{N}_1}, null_\wedge), (SKP_{\mathcal{N}_2}, null_\wedge), (null_\wedge, p_\wedge)\}$
  $colset\ cl = union\ C(SKP_{\mathcal{N}_1}) + C(SKP_{\mathcal{N}_2})$
  $\Sigma_{\mathcal{N}} \leftarrow \Sigma_{\mathcal{N}_1} \cup \Sigma_{\mathcal{N}_2} \cup \{cl\}$
  $C_{\mathcal{N}} \leftarrow C_{\mathcal{N}_1} \cup C_{\mathcal{N}_2} \cup \{(p_\wedge, cl)\}$
return $\mathcal{N}$
end function

▷ The combined relation is $\vee$ (or)
function OR_CPN(ψ_l, ψ_r)
  $\mathcal{N}_1 \leftarrow$ EventFormula_CPN(ψ_l)
  $\mathcal{N}_2 \leftarrow$ EventFormula_CPN(ψ_r)
  $P_{\mathcal{N}} \leftarrow P_{\mathcal{N}_1} \cup P_{\mathcal{N}_2} \cup \{p_{\vee 1}, p_{\vee 2}\}$
  $T_{\mathcal{N}} \leftarrow T_{\mathcal{N}_1} \cup T_{\mathcal{N}_2} \cup \{null_{\vee 1}, null_{\vee 2}, null_{\vee 3}\}$
  $A_{\mathcal{N}} \leftarrow A_{\mathcal{N}_1} \cup A_{\mathcal{N}_2} \cup \{(SKP_{\mathcal{N}_1}, null_{\vee 1}), (SKP_{\mathcal{N}_2}, null_{\vee 2}), (null_{\vee 1}, p_{\vee 1}), (null_{\vee 2}, p_{\vee 1}), (p_{\vee 1}, null_{\vee 3}), (null_{\vee 3}, p_{\vee 2})\}$
  $colset\ cl = union\ C(SKP_{\mathcal{N}_1}) + C(SKP_{\mathcal{N}_2})$
  $\Sigma_{\mathcal{N}} \leftarrow \Sigma_{\mathcal{N}_1} \cup \Sigma_{\mathcal{N}_2} \cup \{cl\}$
  $C_{\mathcal{N}} \leftarrow C_{\mathcal{N}_1} \cup C_{\mathcal{N}_2} \cup \{(p_{\vee 1}, cl), (p_{\vee 2}, cl)\}$
return $\mathcal{N}$
end function

▷ The combined relation is $<$ (before)
function BEFORE_CPN(ψ_l, ψ_r)
  $\mathcal{N}_1 \leftarrow$ EventFormula_CPN(ψ_l)
  $\mathcal{N}_2 \leftarrow$ EventFormula_CPN(ψ_r)
  $n \leftarrow |SCP_{\mathcal{N}_2}|$
  $P' \leftarrow \{p'_1, \ldots, p'_n\}$
  $P_{\mathcal{N}} \leftarrow P_{\mathcal{N}_1} \cup P_{\mathcal{N}_2} \cup P'$
  $T_{\mathcal{N}} \leftarrow T_{\mathcal{N}_1} \cup T_{\mathcal{N}_2} \cup \{null_<\}$
  for all $p \in SCP_{\mathcal{N}_2}$ **do**
    $i \leftarrow 1$
    $A' \leftarrow A' \cup (p'_i, OT_p)$
    $i \leftarrow i + 1$
  end for
  $A_{\mathcal{N}} \leftarrow A_{\mathcal{N}_1} \cup A_{\mathcal{N}_2} \cup \{(SKP_{\mathcal{N}_1}, null_<)\} \cup \{(null_<, p'_1), \ldots, (null_<, p'_n)\} \cup A'$
  $\Sigma_{\mathcal{N}} \leftarrow \Sigma_{\mathcal{N}_1} \cup \Sigma_{\mathcal{N}_2}$
  $C_{\mathcal{N}} \leftarrow C_{\mathcal{N}_1} \cup C_{\mathcal{N}_2} \cup \{(p'_1, C(SKP_{\mathcal{N}_1})), \ldots, (p'_n, C(SKP_{\mathcal{N}_1}))\}$
return $\mathcal{N}$
end function

▷ determine the combined relation
if $CO_\psi = \wedge$ **then**
$\mathcal{N} \leftarrow$ AND_CPN(LT_ψ, RT_ψ)
else if $CO_\psi = \vee$ **then**
$\mathcal{N} \leftarrow$ OR_CPN(LT_ψ, RT_ψ)
else if $CO_\psi = <$ **then**
$\mathcal{N} \leftarrow$ BEFORE_CPN(LT_ψ, RT_ψ)
else
$\mathcal{N} \leftarrow$ Event_CPN(ψ) ▷ Call the algorithm of generating the CPN model of a given event
end if

A.4 Mapping from a regulative norm to CPNs

Algorithm 11 gives the procedure of generating the CPN model of a given regulative norm.

Algorithm 11 CPN mapping of a regulative norm: RN_CPN

Require: (n^r, REA) ▷ A regulative norm and a color set for the list of the union of all roles in the normative system
Ensure: $(\mathcal{N}, P_s, P_v, P_{vr}, p_{so}, p_{vo})$ ▷ A CPN mapping of NN
$(D, \rho, \delta, \sigma) \leftarrow n^r$

▷ (1) Construct the CPN model of the target ρ, deadline δ and precondition σ
$\mathcal{N}_\rho \leftarrow$ EventFormula_CPN(ρ)
$\mathcal{N}_\delta \leftarrow$ EventFormula_CPN(δ)
$\mathcal{N}_\sigma \leftarrow$ EventFormula_CPN(σ)

▷ (2) Distinguish the satisfied and violated states according to the deontic type of the regulative norm
if $D =$ O **then**
$(P_{so}, P_{vo}) \leftarrow (SKP_{\mathcal{N}_\rho}, SKP_{\mathcal{N}_\delta})$
end if
if $D =$ F **then**
$(P_{so}, P_{vo}) \leftarrow (SKP_{\mathcal{N}_\rho}, SKP_{\mathcal{N}_\delta})$
end if
$(P_s, P_v) \leftarrow (P_{so}, P_{vo})$

▷ (3) Connect the CPN models of the target, deadline and precondition
$P_\mathcal{N} \leftarrow P_{\mathcal{N}_\rho} \cup P_{\mathcal{N}_\delta} \cup P_{\mathcal{N}_\sigma}$
$T_\mathcal{N} \leftarrow T_{\mathcal{N}_\rho} \cup T_{\mathcal{N}_\delta} \cup T_{\mathcal{N}_\sigma}$
$A_\mathcal{N} \leftarrow A_{\mathcal{N}_\rho} \cup A_{\mathcal{N}_\delta} \cup A_{\mathcal{N}_\sigma} \cup \{(SCP_{\mathcal{N}_\rho}, IT_{SKP_{\mathcal{N}_\delta}}), (SKP_{\mathcal{N}_\sigma}, IT_{SKP_{\mathcal{N}_\rho}}), (SKP_{\mathcal{N}_\sigma}, IT_{SKP_{\mathcal{N}_\delta}})\}$

$\Sigma_{\mathcal{N}} \leftarrow \Sigma_{\mathcal{N}_\rho} \cup \Sigma_{\mathcal{N}_\delta} \cup \Sigma_{\mathcal{N}_\sigma}$
$(C(SKP_{\mathcal{N}_\rho}), C(SKP_{\mathcal{N}_\delta})) \leftarrow (REA, REA)$
$C_{\mathcal{N}} \leftarrow C_{\mathcal{N}_\rho} \cup C_{\mathcal{N}_\delta} \cup C_{\mathcal{N}_\sigma}$

A.5 Mapping from a regulative norm net to CPNs

Algorithm 12 gives the procedure of generating the CPN model of a given regulative norm net.

Algorithm 12 CPN mapping of a regulative norm net: RNN_CPN

Require: RNN ▷ A regulative norm net
Ensure: $(\mathcal{N}, P_s, P_v, P_{vr}, p_{so}, p_{vo})$ ▷ A CPN mapping of regulative norm net

▷ The compliance relation is AND
function AND_CPN(RNN_l, RNN_r, REA)
  $(\mathcal{N}_1, P_{s1}, P_{v1}, P_{vr1}, p_{so1}, p_{vo1}) \leftarrow$ RNN_REA(RNN_l)
  $(\mathcal{N}_2, P_{s2}, P_{v2}, P_{vr2}, p_{so2}, p_{vo2}) \leftarrow$ RNN_REA(RNN_r)
  $P_{\mathcal{N}} \leftarrow P_{\mathcal{N}_1} \cup P_{\mathcal{N}_2} \cup \{p_s, p_v\}$
  $T_{\mathcal{N}} \leftarrow T_{\mathcal{N}_1} \cup T_{\mathcal{N}_2} \cup \{null_s, null_{v1}, null_{v2}\}$
  $A_{\mathcal{N}} \leftarrow A_{\mathcal{N}_1} \cup A_{\mathcal{N}_2} \cup \{(p_{so1}, null_s), (p_{so2}, null_s), (p_{vo1}, null_{v1}), (p_{vo2}, null_{v2}), (null_s, p_s), (null_{v1}, p_v), (null_{v2}, p_v)\}$
  $\Sigma_{\mathcal{N}} \leftarrow \Sigma_{\mathcal{N}_1} \cup \Sigma_{\mathcal{N}_2}$
  $C_{\mathcal{N}} \leftarrow C_{\mathcal{N}_1} \cup C_{\mathcal{N}_2} \cup \{(p_s, REA), (p_v, REA)\}$
  $(P_s, P_v, P_{vr}, p_{so}, p_{vo}) \leftarrow (P_{s1} \cup P_{s2} \cup \{p_s\}, P_{v1} \cup P_{v2} \cup \{p_v\}, P_{vr1} \cup P_{vr2}, p_s, p_v)$
return $(\mathcal{N}, P_s, P_v, P_{vr}, p_{so}, p_{vo})$
end function

▷ The compliance relation is OR
function OR_CPN(RNN_l, RNN_r, REA)
  $(\mathcal{N}_1, P_{s1}, P_{v1}, P_{vr1}, p_{so1}, p_{vo1}) \leftarrow$ RNN_REA(RNN_l)
  $(\mathcal{N}_2, P_{s2}, P_{v2}, P_{vr2}, p_{so2}, p_{vo2}) \leftarrow$ RNN_REA(RNN_r)
  $P_{\mathcal{N}} \leftarrow P_{\mathcal{N}_1} \cup P_{\mathcal{N}_2} \cup \{p_s, p_v\}$
  $T_{\mathcal{N}} \leftarrow T_{\mathcal{N}_1} \cup T_{\mathcal{N}_2} \cup \{null_{s1}, null_{s2}, null_v\}$
  $A_{\mathcal{N}} \leftarrow A_{\mathcal{N}_1} \cup A_{\mathcal{N}_2} \cup \{(p_{s1}, null_{s1}), (p_{s2}, null_{s2}), (p_{v1}, null_v), (p_{v2}, null_v), (null_{s1}, p_s), (null_{s2}, p_s), (null_v, p_v)\}$
  $\Sigma_{\mathcal{N}} \leftarrow \Sigma_{\mathcal{N}_1} \cup \Sigma_{\mathcal{N}_2}$
  $C_{\mathcal{N}} \leftarrow C_{\mathcal{N}_1} \cup C_{\mathcal{N}_2} \cup \{(p_s, REA), (p_v, REA)\}$
  $(P_s, P_v, P_{vr}, p_{so}, p_{vo}) \leftarrow (P_{s1} \cup P_{s2} \cup \{p_s\}, P_{v1} \cup P_{v2} \cup \{p_v\}, P_{vr1} \cup P_{vr2}, p_s, p_v)$
return $(\mathcal{N}, P_s, P_v, P_{vr}, p_{so}, p_{vo})$
end function

▷ The compliance relation is OE
function OE_CPN(RNN_l, RNN_r, REA)
    $(\mathcal{N}_1, P_{s1}, P_{v1}, P_{vr1}, p_{so1}, p_{vo1}) \leftarrow$ RNN_REA(RNN_l)
    $(\mathcal{N}_2, P_{s2}, P_{v2}, P_{vr2}, p_{so2}, p_{vo2}) \leftarrow$ RNN_REA(RNN_r)
    $P_{\mathcal{N}} \leftarrow P_{\mathcal{N}_1} \cup P_{\mathcal{N}_2} \cup \{p_{vr}, p_s, p_v\}$
    $T_{\mathcal{N}} \leftarrow T_{\mathcal{N}_1} \cup T_{\mathcal{N}_2} \cup \{null_{s1}, null_{v1}, null_{s2}, null_{v2}\}$
    for all $p \in SCP_{\mathcal{N}_2}$ **do**
        $A' \leftarrow A' \cup (p_{vr}, OT_p)$
    end for
    $A_{\mathcal{N}} \leftarrow A_{\mathcal{N}_1} \cup A_{\mathcal{N}_2} \cup A' \cup \{(p_{s1}, null_{s1}), (p_{v1}, null_{v1}), (null_{v1}, p_{vr}), (p_{s2}, null_{s2}), (p_{v2}, null_{v2}), (null_{s1}, p_s), (null_{s2}, p_s), (null_{v2}, p_v)\}$
    $\Sigma_{\mathcal{N}} \leftarrow \Sigma_{\mathcal{N}_1} \cup \Sigma_{\mathcal{N}_2}$
    $C_{\mathcal{N}} \leftarrow C_{\mathcal{N}_1} \cup C_{\mathcal{N}_2} \cup \{(p_s, REA), (p_v, REA), (p_{vr}, REA)\}$
    $(P_s, P_v, P_{vr}, p_{so}, p_{vo}) \leftarrow (P_{s1} \cup P_{s2} \cup \{p_s\}, P_{v1} \cup P_{v2} \cup \{p_v\}, P_{vr1} \cup P_{vr2} \cup \{p_{vr}\}, p_s, p_v)$
return $(\mathcal{N}, P_s, P_v, P_{vr}, p_{so}, p_{vo})$
end function

▷ **(1)** Create a color set representing all the role enacting agents that are regulated by the norms defined in the regulative norm net RNN
$\{r_1, \ldots, r_n\} \leftarrow$ getRoles(RNN)
for all $i = 1 \rightarrow n$ **do**
    $colset\ cr_i = with\ name_{r_i}$
end for
$colset\ rea = union\ cr_1 + \ldots + cr_n$
$colset\ REA = list\ rea$

▷ **(2)** Determine the compliance relation of a given RNN and obtain its CPN mapping
function RNN_REA(RNN, REA)
    if CR_{RNN} = AND **then**
        $(\mathcal{N}, P_s, P_v, P_{vr}, p_{so}, p_{vo}) \leftarrow$ AND_CPN(LT_{RNN}, RT_{RNN}, REA)
    else if CR_{RNN} = OR **then**
        $(\mathcal{N}, P_s, P_v, P_{vr}, p_{so}, p_{vo}) \leftarrow$ OR_CPN(LT_{RNN}, RT_{RNN}, REA)
    else if CR_{RNN} = OE **then**
        $(\mathcal{N}, P_s, P_v, P_{vr}, p_{so}, p_{vo}) \leftarrow$ OE_CPN(LT_{RNN}, RT_{RNN}, REA)
    else
        $(\mathcal{N}, P_s, P_v, P_{vr}, p_{so}, p_{vo}) \leftarrow$ RN_CPN(RNN, REA)
    end if
end function

A.6 Mapping from a constitutive norm to CPNs

Algorithm 13 gives the procedure of generating the CPN model of a given constitutive norm.

Algorithm 13 CPN Mapping of a constitutive norm: CN_CPN

Require: n^c ▷ A constitutive norm
Ensure: $\mathcal{N}_H$ ▷ A hierarchical CPN model

$(\beta, \kappa, \nu) \leftarrow n^r$
▷ Construct a CPN module based on the institutional event ν
$\mathcal{N}_\nu \leftarrow$ Event_CPN(ν)
$T_{sub} \leftarrow T_{\mathcal{N}_\nu}$
$\mathcal{N}_{PM} \leftarrow (\mathcal{N}_\nu, T_{sub}, \emptyset, \emptyset)$
▷ Construct a CPN module based on the external events β and κ
$(r, \alpha) \leftarrow \nu$
$\mathcal{N}_a \leftarrow$ Event_CPN($r, null$)
$\mathcal{N}_b \leftarrow$ EventFormula_CPN($\kappa < \beta$)
$\mathcal{N}_c \leftarrow (P_{\mathcal{N}_a} \cup P_{\mathcal{N}_b}, T_{\mathcal{N}_a} \cup T_{\mathcal{N}_b}, A_{\mathcal{N}_a} \cup A_{\mathcal{N}_b} \cup (SKP_{\mathcal{N}_b}, OT_{SCP_{\mathcal{N}_a}}), \Sigma_{\mathcal{N}_a} \cup \Sigma_{\mathcal{N}_b}, C_{\mathcal{N}_a} \cup C_{\mathcal{N}_b}$
$P_{port} \leftarrow \{SCP_{\mathcal{N}_a}, SKP_{\mathcal{N}_a}\}$
$PT \leftarrow \{(SCP_{\mathcal{N}_a}, IN), (SKP_{\mathcal{N}_a}, OUT)\}$
$\mathcal{N}_{SM} \leftarrow (\mathcal{N}_c, \emptyset, P_{port}, PT)$
▷ Construct a hierarchical CPN based on the counts-as relation between the external events β, κ and the institutional event ν
$S \leftarrow \{\mathcal{N}_{PM}, \mathcal{N}_{SM}\}$
$SM \leftarrow (T_{\mathcal{N}_\nu}, \mathcal{N}_{SM})$
$PS \leftarrow \{(SCP_{\mathcal{N}_{PM}}, SCP_{\mathcal{N}_a}), (SKP_{\mathcal{N}_{PM}}, SKP_{\mathcal{N}_a})\}$
$\mathcal{N}_H \leftarrow (S, SM, PS)$

A.7 Norm Compliance Checking

Algorithm 14 gives the procedure of checking the compliance status of a norm net with respect to a given interaction plan.

Algorithm 14 Compliance Query: CLQ

Require: (NN^i, IP) ▷ A norm net instance and an interaction plan
Ensure: CE ▷ Compliance evaluation results

▷ Obtain the new marking of a CPN model $\mathcal{N}$ given the occurrence of an enabled step Y with the current marking M
function UPDATESTATE($Y, M, \mathcal{N}$)
  for all $p \in P_{\mathcal{N}}$ **do**
    $M'(p) \leftarrow (M(p) -- \overset{++}{\underset{MS}{}} \sum_{(t,b)\in Y} E(p,t)\langle b\rangle) ++ \overset{++}{\underset{MS}{}} \sum_{(t,b)\in Y} E(t,p)\langle b\rangle$
  end for
  if EStep$((*, null), \mathcal{N}, M') \neq \emptyset$ **then**
    $M' \leftarrow$ UpdateState($EStep((*, null), \mathcal{N}, M'), M', \mathcal{N}$)
  end if
return M'
end function

▷ Evaluate the compliance evaluation state given two normative states
function COMPLIANCEEVA(M, M', P_{vr}, p_{vo})
  if $M'(p_{vo}) - -M(p_{vo}) \neq \emptyset$ **then**
    $ce \leftarrow$ *violated*
  else
    $ce \leftarrow$ *compliant*
  end if
  if ce = *compliant* **then**
    for all $p \in P_{vr}$ **do**
      if $M'(p) - -M(p) \neq \emptyset$ **then**
        $ce \leftarrow$ *sub-ideal compliant*
      end if
    end for
  end if
return ce
end function

▷ (**Step 1**) Obtain the CPN mapping of the norm net instance NN^i
$\langle(\mathcal{N}, P_s, P_v, P_{vr}, p_{so}, p_{vo}), M\rangle \leftarrow \Theta(NN^i)$

▷ (**Step 2**) Obtain the changes of the compliance evaluation state of NN^i with respect to the interaction plan IP
for $j \leftarrow 1$ to $|IP|$ **do**
  ▷ (**Step 2.1**) Obtain the new normative state of NN^i given the occurrence of the event $IP[j]$
  $M' \leftarrow$ UpdateState(EStep($IP[j], \mathcal{N}, M$), $M, \mathcal{N}$)
  ▷ (**Step 2.2**) Evaluate the compliance status of NN^i with respect to the occurrence of the event $IP[j]$
  $(CE[j]) \leftarrow$ ComplianceEva(M, M', P_{vr}, p_{vo})
  $M \leftarrow M'$
end for

A.8 Norm Conflict Detection

Algorithm 15 gives the procedure of detecting norm conflicts in a norm net instance with respect to the occurrence of an external event.

Algorithm 15 Conflict Query: CFQ

Require: (NN^i, e) ▷ A norm net instance and an external event
Ensure: CFS ▷ Conflicting status

▷ Obtain the new marking of a CPN model $\mathcal{N}$ given the occurrence of an enabled step Y with the current marking M
function UPDATESTATE($Y, M, \mathcal{N}$)
  for all $p \in P_{\mathcal{N}}$ **do**
    $M'(p) \leftarrow (M(p) -- \overset{++}{MS} \sum_{(t,b)\in Y} E(p,t)\langle b\rangle) ++ \overset{++}{MS} \sum_{(t,b)\in Y} E(t,p)\langle b\rangle$
  end for
  if EStep$((*, null), \mathcal{N}, M') \neq \emptyset$ **then**
    $M' \leftarrow$ UpdateState(EStep$((*, null), \mathcal{N}, M'), M', \mathcal{N})$
  end if
return M'
end function

▷ (**Step 1**) Obtain the CPN mapping of the norm net instance NN^i
$\langle(\mathcal{N}, P_s, P_v, P_{vr}, p_{so}, p_{vo}), M\rangle \leftarrow \Theta(NN^i)$

▷ (**Step 2**) Obtain the new normative state of NN^i after the event e occurs
$M' \leftarrow$ UpdateState(EStep$(e, \mathcal{N}, M), M, \mathcal{N})$

▷ (**Step 3**) Check the normative state changes of all the component regulative norm nets in NN^i
  ▷ (**Step 3.1**) Check whether the normative state of NN^i is evaluated to be violated with respect to the occurrence of the event e
$CFS \leftarrow false$
if $M'(p_{vo}) - -M(p_{vo}) > 0$ **then**
  for all $(p_s, p_v) \in P_s \times P_v$ **do**
    ▷ (**Step 3.2**) Check whether there are two component regulative norm nets in NN^i such that one is evaluated to be satisfied and the other is evaluated to be violated with respect to the occurrence of the event e
    if $(M'(p_s){-}{-}M(p_s)) > 0$ **and** $(M'(p_v){-}{-}M(p_v)) > 0$ **then**
      $CFS \leftarrow true$
    end if
  end for
end if

A.9 Combined Evaluation of Norm Compliance and Preference Satisfaction

Algorithm 16 gives the procedure of checking the compliance status of a norm net and the satisfaction status of a set of agents with respect to a given interaction plan.

Algorithm 16 Evaluation of norm compliance and preference satisfaction

---

Require: $(\langle(\mathcal{N}_H, P_s, P_v, P_{vr}, p_{so}, p_{vo}, P_{PS}), M\rangle, IP)$
Ensure: (CE, PS)

▷ Obtain the new marking of a CPN model $\mathcal{N}$ given the occurrence of an enabled step Y with the current marking M
function UPDATESTATE$(Y, M, \mathcal{N})$
  for all $p \in P_{\mathcal{N}}$ **do**
    $M'(p) \leftarrow (M(p) -- \overset{++}{MS} \sum_{(t,b)\in Y} E(p,t)\langle b\rangle) ++ \overset{++}{MS} \sum_{(t,b)\in Y} E(t,p)\langle b\rangle$
  end for
  if $EStep((*, null), \mathcal{N}, M') \neq \emptyset$ **then**
    $M' \leftarrow$ UpdateState(EStep($(*, null), \mathcal{N}, M'$), $M', \mathcal{N}$)
  end if
  if $EStep((*, min), \mathcal{N}, M') \neq \emptyset$ **then**
    $M' \leftarrow$ UpdateState(EStep($(*, min), \mathcal{N}, M'$), $M', \mathcal{N}$)
  end if
  if $EStep((*, sum), \mathcal{N}, M') \neq \emptyset$ **then**
    $M' \leftarrow$ UpdateState(EStep($(*, sum), \mathcal{N}, M'$), $M', \mathcal{N}$)
  end if
return M'
end function

▷ Norm compliance and preference satisfaction
for $j = 1$ to $|IP|$ **do**
  $M' \leftarrow$ UpdateMarking(ESteps($IP[j], \mathcal{N}, M$), $M, \mathcal{N}_H$)
  ▷ (1) Compliance checking
  $CE[j] \leftarrow$ ComplianceEva(M, M', P_{vr}, p_{vo})
end for
▷ (2) Satisfaction evaluation
for all $p \in P_{\mathrm{PS}}$ **do**
  $i \leftarrow 1$
  $PS[i] \leftarrow M'(p)$
  $i \leftarrow i + 1$
end for

Bibliography

[1] W. M. P. van der Aalst. *Process Mining: Discovery, Conformance and Enhancement of Business Processes*. Springer, 2011.

[2] W. M. P. van der Aalst. Process mining: Overview and opportunities. *ACM Transaction on Management Information Systems*, 3(2):pages 1–17, 2012.

[3] W. M. P. van der Aalst, A. Adriansyah, and B. F. van Dongen. Replaying history on process models for conformance checking and performance analysis. *Data Mining and Knowledge Discovery*, 2(2):pages 182–192, 2012.

[4] W. M. P. van der Aalst, K. van Hee, J. M. van der Werf, A. Kumar, and M. Verdonk. Conceptual model for online auditing. *Decision Support Systems*, 50(3):pages 636–647, 2011.

[5] T. Ågotnes, W. van der Hoek, and M. Wooldridge. Robust normative systems and a logic of norm compliance. *Logic Journal of the IGPL*, 18(1):pages 4–30, 2010.

[6] C. E. Alchourrón and E. Bulygin. *Normative Systems*. Springer Verlag, 1971.

[7] H. Aldewereld. *Autonomy vs. Conformity: an Institutional Perspective on Norms and Protocols*. Ph.D. thesis, 2007.

[8] H. Aldewereld, S. Alvarez-Napagao, F. Dignum, and J. Vázquez-Salceda. Engineering social reality with inheritence relations. In *Proceedings of the International Workshop on Engineering Societies in the Agents World X*, pages 116–131. 2009.

[9] H. Aldewereld, S. Álvarez-Napagao, F. Dignum, and J. Vázquez-Salceda. Making norms concrete. In *Proceedings of the 9th International Conference on Autonomous Agents and Multiagent Systems*, pages 807–814. 2010.

[10] H. Aldewereld and V. Dignum. OperettA: Organization-oriented development environment. In *Languages, Methodologies, and Development Tools for Multi-Agent Systems*, pages 1–18. 2011.

[11] H. Aldewereld, J. Vázquez-Salceda, F. Dignum, and J.-J. Ch. Meyer. Verifying norm compliancy of protocols. In *Coordination, Organizations, Institutions, and Norms in Multi-Agent Systems*, pages 231–245. 2006.

[12] T. Allweyer. *BPMN 2.0: Introduction to the Standard for Business Process Modeling*. Books on Demand GmbH, 2010.

[13] G. Andrighetto, G. Governatori, P. Noriega, and L. van der Torre. *Normative Multi-Agent Systems*. Schloss Dagstuhl, 2012.

[14] E. Argente. *GORMAS: Guías para el desarrollo de Sistemas Multiagente abiertos basados en organizaciones*. Ph.D. thesis, Universitat Politècnica de València, 2008.

[15] L. Astefanoaei, M. Dastani, J.-J. Ch. Meyer, and F. S. de Boer. On the semantics and verification of normative multi-agent systems. *International Journal of Universal Computer Science*, 15(13):pages 2629–2652, 2009.

[16] A. Awad, R. Goré, Z. Hou, J. Thomson, and M. Weidlich. An iterative approach to synthesize business process templates from compliance rules. *Information Systems*, 37(8):pages 714–736, 2012.

[17] C. W. Bachman and M. Daya. The role concept in data models. In *Proceedings of the Third International Conference on Very Large Data Bases - Volume 3*, pages 464–476. 1977.

[18] C. I. Barnard. *The Functions of the Executive*. Harvard University Press, 1968.

[19] P. L. Berger and T. Luckmann. *The social construction of reality*. New York: Doubleday Anchor, 1967.

[20] B. J. Biddle. *Role Theory: Expectations, Identities, and Behaviors*. Academic Press, 1979.

[21] B. J. Biddle. Recent development in role theory. *Annual Review of Sociology*, 12:pages 67–92, 1986.

[22] B. J. Biddle and E. J. Thomas. *Role Theory: Concepts and Research*. Wiley, 1966.

[23] M. Bienvenu, C. Fritz, and S. A. McIlraith. Specifying and computing preferred plans. *Artificial Intelligence*, 175(7–8):pages 1308–1345, 2011.

[24] Binder Dijker Otte Co. The consequences of non-compliance in global business. Technical report, Binder Dijker Otte & Co., 2011.

[25] G. Boella, G. Pigozzi, and L. van der Torre. Normative systems in computer science. Ten guidelines for normative multiagent systems. In *Normative Multi-Agent Systems, Dagstuhl Seminar Proceedings*. 2009.

[26] G. Boella and F. Steimann. Roles and relationships in object-oriented programming, multiagent systems and ontologies. In *Proceedings of ECOOP 2007 Workshop Reader*, pages 108–122. 2008.

[27] G. Boella and L. van der Torre. Organizations as socially constructed agents in the agent oriented paradigm. In *ESAW*, pages 1–13. 2004.

[28] G. Boella and L. van der Torre. Regulative and constitutive norms in normative multiagent systems. In *Proceedings of the 10th International Conference on the Principles of Knowledge Representation and Reasoning*, pages 255–266. 2004.

[29] G. Boella and L. van der Torre. Constitutive norms in the design of normative multiagent systems. In *CLIMA VI*, pages 303–319. 2006.

[30] R. H. Bordini, J. F. Hübner, and M. Wooldrige. *Programming Multi-Agent Systems in AgentSpeak using Jason*. John Wiley & Sons, 2007.

[31] A. Braganza and A. Franken. SOX, compliance, and power relationships. *Communications of the ACM*, 50(9):pages 97–102, 2007.

[32] M. E. Bratman. *Intention, plans, and practical reason*. Harvard University Press, 1987.

[33] M. E. Bratman, D. J. Israel, and M. E. Pollack. Plans and resource-bounded practical reasoning. *Computational Intelligence*, 4(4):pages 349–355, 1988.

[34] P. Bresciani, A. Perini, P. Giorgini, F. Giunchiglia, and J. Mylopoulos. Tropos: An agent-oriented software development methodology. *Journal of Autonomous Agents and Multi-Agent Systems*, 8:pages 203–236, 2004.

[35] J. Broersen, M. Dastani, J. Hulstijn, and L. van der Torre. Goal generation in the BOID architecture. *Cognitive Science Quarterly*, 2(3–4):pages 428–447, 2002.

[36] R. Brooks. A robust layered control system for a mobile robot. *Journal of Robotics and Automation*, 2(1):pages 14–23, 1986.

[37] N. Bulling, M. Dastani, and M. Knobbout. Monitoring norm violations in multi-agent systems. In *Proceedings of the 12th international conference on Autonomous agents and multiagent systems*, pages 491–498. 2013.

[38] B. Burgemeestre, J. Hulstijn, and Y. H. Tan. Towards an architecture for self-regulating agents: A case study in international trade. In *Coordination, Organizations, Institutions and Norms in Agent Systems V*, pages 320–333. 2009.

[39] L. Cabac, T. Dörges, and H. Rölke. A monitoring toolset for Paose. In *Applications and Theory of Petri Nets*, pages 399–408. 2008.

[40] F. Caron, J. Vanthienen, and B. Baesens. Comprehensive rule-based compliance checking and risk management with process mining. *Decision Support Systems*, 54(3):pages 1357–1369, 2013.

[41] C. Castelfranchi. Formalising the informal? Dynamic social order, bottom-up social control, and spontaneous normative relations. *Journal of Applied Logic*, 1(1–2):pages 47–92, 2003.

[42] J. R. Celaya, A. A. Desrochers, and R. J. Graves. Modeling and analysis of multi-agent systems using petri nets. *Journal of computers*, 4(10):pages 981–996, 2009.

[43] O. Cliffe, M. De Vos, and J. Padget. Specifying and reasoning about multiple institutions. In *Coordination, Organizations, Institutions, and Norms in Agent Systems II*, pages 67–85. 2007.

[44] J. S. Coleman. *Power and the structure of society*. W. W. Norton & Co., Inc, 1974.

[45] D. Corapi, A. Russo, M. De Vos, J. Padget, and K. Satoh. Normative design using inductive learning. *Theory and Practice of Logic Programming*, 11:pages 783–799, 2011.

[46] M. Cossentino, N. Gaud, V. Hilaire, S. Galland, and A. Koukam. ASPECS: an agent-oriented software process for engineering complex systems. *Journal of Autonomous Agents and Multi-Agent Systems*, 20:pages 260–304, 2010.

[47] D. D'Aprile, L. Giordano, V. Gliozzi, A. Martelli, G. L. Pozzato, and D. T. Dupré. Verifying business process compliance by reasoning about actions. In *Proceedings of the International Workshop on Computational Logic in Multi-Agent Systems XI*, pages 99–116. 2010.

[48] A. Dasgupta and A. K. Ghose. BDI agents with objectives and preferences. In *Proceeding of Declarative Agent Languages and Technologies VIII*, pages 22–39. 2010.

[49] M. Dastani, D. Grossi, J.-J. Ch. Meyer, and N. Tinnemeier. Normative multi-agent programs and their logics. In *Proceedings of the Workshop on Knowledge Representation for Agents and Multi-Agent Systems*, pages 16–31. 2009.

[50] J. Delgrande, T. Schaub, and H. Tompits. Domain-specific preferences for causual reasoning and planning. In *Proceedings of the International conference on knowledge representation and reasoning*, pages 673–682. 2004.

[51] S. A. DeLoach. The MaSE methodology. In *Methodologies and Software Engineering for Agent Systems*, pages 107–125. 2004.

[52] A. K. Dey. Understanding and using context. *Personal Ubiquitous Computing*, 5(1):pages 4–7, 2001.

[53] F. Dignum. Abstract norms and electronic institutions. In *Proceedings of the International Workshop on Regulated Agent-Based Social Systems: Theories and Applications*, pages 93–104. 2002.

[54] F. Dignum, H. Weigand, and E. Verharen. Meeting the deadline: On the formal specification of temporal deontic constraints. In *Foundations of Intelligent Systems*, pages 243–252. 1996.

[55] V. Dignum. *A model for organizational interaction: based on agents, founded in logic*. Ph.D. thesis, Utrecht University, 2004.

[56] V. Dignum, editor. *Handbook of Research on Multi-Agent Systems: Semantics and Dynamics of organizational Models*. Information Science Reference, 2009.

[57] L. B. Edelman and A. Talesh, Shauhin. *Explaining Compliance: Business Responses to Regulation*. Edward Elgar Pub, 2011.

[58] M. Esteva, J. A. Rodríguez-Aguilar, B. Rosell, and J. L. Arcos. AMELI: An agent-based middleware for electronic institutions. In *Proceedings of the third international joint conference on autonomous agents and multi-agent systems*, pages 236–243. 2004.

[59] European Commission. Authorised economic operator. http://ec.europa.eu/taxation_customs/customs/policy_issues/customs_security/aeo/, 2013.

[60] European Commission. The community customs code, implementing provisions and guidelines. http://ec.europa.eu/taxation_customs/customs/procedural_aspects/general/community_code/, 2013.

[61] European Commission. Customs declaration. http://ec.europa.eu/taxation_customs/customs/procedural_aspects/general/declaration/index_en.htm, 2013.

[62] European Commission. Entry summary declaration. http://eur-lex.europa.eu/LexUriServ/LexUriServ.do?uri=CONSLEG:1993R2454:20090701:en:PDF, 2013.

[63] M. Fellmann and A. Zasada. State-of-the-art of business process compliance approach: a survey. In *Proceedings of the 22nd European Conference on Information Systems*. 2014.

[64] S. Fenech, G. J. Pace, and G. Schneider. Automatic conflict detection on contracts. In *Proceedings of the 6th International Colloquium on Theoretical Aspects of Computing*, pages 200–214. 2009.

[65] J. Ferber and O. Gutknecht. A meta-model for the analysis and design of organizations in multi-agent systems. In *Proceedings of the 3rd International Conference on Multi Agent Systems*, pages 128–135. 1998.

[66] J. Ferber, O. Gutknecht, and F. Michel. From agents to organizations: An organizational view of multi-agent systems. In *Agent-Oriented Software Engineering IV*, pages 214–230. 2003.

[67] I. Fergusson. Integrated control and coordinated behavior: A case for agent models. In *Intelligent Agent: Theories, Architectures and Languages*, pages 203–218. 1995.

[68] R. Foorthuis and R. Bos. A framework for organizational compliance management tactics. In *Proceeding of Advanced Information Systems Engineering Workshops*, pages 259–268. 2011.

[69] N. Fornara. Specifying and monitoring obligations in open multiagent systems using semantic web technology. In *Semantic Agent Systems: Foundations and Applications*, pages 25–46. 2011.

[70] N. Fornara and M. Colombetti. Specifying artificial institutions in the event calculus. In *Handbook of Research on Multi-Agent Systems: Semantics and Dynamics of Organizational Models*, pages 335–366. 2009.

[71] M. Fowler and K. Scott. *UML Distilled Second Edition - A Brief Guide to the Standard Object Modeling Language*. Object Technology Seriese: Addison-Wesley, 1999.

[72] D. Gaertner, A. Garcia-Camino, P. Noriega, J. A. Rodriguez-Aguilar, and W. Vasconcelos. Distributed norm management in regulated multi-agent systems. In *Proceedings of the International Conference on Autonomous Agents and Multiagent Systems*, pages 624–631. 2007.

[73] X. Gao and M. P. Singh. Extracting normative relationships from business contracts. In *Proceedings of the 2014 International Conference on Autonomous Agents and Multi-agent Systems*, pages 101–108. 2014.

[74] E. Garcia, A. Giret, and V. Botti. Developing regulated open multi-agent systems. In *Proceedings of the International Conference on Agreement Technologies*, pages 12–26. 2012.

[75] E. Garcia, A. Giret, and V. Botti. A model-driven CASE tool for developing and verifying regulated open MAS. *Science of Computer Programming*, 78(6):pages 695–704, 2013.

[76] D. Gauthier. Contractarianism and rational choice: Essays on david gauthier's morals by agreement. Cambridge University Press, 1991.

[77] J. Gelati, A. Rotolo, G. Sartor, and G. Governatori. Normative autonomy and normative co-ordination: Declarative power, representation, and mandate. *Artificial Intelligence and Law*, 12(1–2):pages 53–81, 2004.

[78] M. Genesereth and N. Nilsson. *Logical Foundations of Artificial Intelligence*. Morgan Kaufman, 1987.

[79] G. K. Giannikis and A. Daskalopulu. Normative conflicts in electronic contracts. *Electronic Commerce Research and Applications*, 10(2):pages 247–267, 2011.

[80] F. Giunchiglia and P. Bouquet. Introduction to contextual reasoning. An artificial intelligence perspective. In *Proceeding of Perspectives on Cognitive Science*, pages 138–157. 1996.

[81] S. Goedertier and J. Vanthienen. Designing compliant business processes with obligations and permissions. In *Proceedings of the Business Process Management Workshops*, pages 5–14. 2006.

[82] E. Goffman. *The Presentation of Self in Everyday Life*. Double Day Anchor Books, Garden City, 1959.

[83] G. Governatori, J. Gelati, A. Rotolo, and G. Sartor. Actions, institutions, powers. preliminary notes. In *Proceedings of the International Workshop on Regulated Agent-Based Social Systems: Theories and Applications*, pages 131–147. 2002.

[84] G. Governatori, J. Hulstijn, R. Riveret, and A. Rotolo. Characterising deadlines in temporal modal defeasible logic. In *Australian Conference on Artificial Intelligence*, volume 4830, pages 486–496. 2007.

[85] G. Governatori and Z. Milosevic. A formal analysis of a business contract language. *International Journal of Cooperative Information Systems*, 15(4):pages 659–685, 2006.

[86] G. Governatori and D. H. Pham. r-contract: An architecture for e-contracts in defeasible logic. *International Journal of Business Process Integration and Management*, 4(3):pages 187–199, 2009.

[87] G. Governatori and A. Rotolo. An algorithm for business process compliance. In *Proceedings of the International Conference on Legal Knowledge and Information Systems*, pages 186–191. 2008.

[88] G. Governatori and A. Rotolo. How do agents comply with norms? In *Proceedings of the International Joint Conferences on Web Intelligence and Intelligent Agent Technologies*, pages 488–491. 2009.

[89] G. Governatori and A. Rotolo. A conceptually rich model of business process compliance. In *APCCM*, pages 3–12. 2010.

[90] G. Governatori and A. Rotolo. Norm compliance in business process modeling. In *Proceedings of the 4th International Web Rule Symposium*, pages 194–209. 2010.

[91] D. Grossi, H. Aldewereld, J. Vazquez-Salceda, and F. Dignum. Ontological aspects of the implementation of norms in agent-based electronic institutions. *Computational and Mathematical Organization Theory*, 12(2-3):pages 251–275, 2006.

[92] D. Grossi, F. Dignum, and J.-J. Ch. Meyer. Context in categorization. In *Proceedings of the Workshop on Context Representation and Reasoning*. 2005.

[93] D. Grossi, F. Dignum, and J.-J. Ch. Meyer. Contextual terminologies. In *Proceedings of the 6th International Workshop on Computational Logic in Multi-Agent Systems*, pages 284–302. 2006.

[94] D. Grossi, J.-J. Ch. Meyer, and F. Dignum. Classificatory aspects of counts-as: An analysis in modal logic. *Journal of Logic and Computation*, 16(5):pages 613–643, 2006.

[95] D. Grossi, J.-J. Ch. Meyer, and F. Dignum. The many faces of counts-as: A formal analysis of constitutive-rules. *Journal of Applied Logic.*, 6(2):pages 192–217, 2008.

[96] N. Guarino. Concepts, attributes and arbitrary relations. *Data & Knowledge Engineering*, 8(3):pages 249–261, 1992.

[97] O. Gutknecht and J. Ferber. The MadKit agent platform architecture. In *Infrastructure for Agents*, pages 48–55. 2001.

[98] M. Hack. *Decidability questions for Petri Nets*. Ph.D. thesis, 1976.

[99] J. Harris and M. Cummings. Compliance issues and IS degree programs. *Journal of Computing Sciences in Colleges*, 23(1):pages 14–20, 2007.

[100] H. L. A. Hart. *The Concept of Law*. Clarendon Press, Oxford, UK, 1961.

[101] E. M. Hartman. *Organizational Ethics and the Good Life*. Oxford University Press, 1995.

[102] S. Henningsson, N. Bjørn-Andersen, A. Schmidt, B. Flügge, and Z. Henriksen. Food living lab – complexity of export trade. In Y. H. Tan, N. Bjørn-Andersen, S. Klein, and B. Rukanova, editors, *Accelerating Global Supply Chains with IT-Innovation*. Berlin: Springer-Verlag, 2011.

[103] A. Herzig, E. Lorini, F. Moisan, and N. Troquard. A dynamic logic of normative systems. In *Proceedings of the International Joint Conference on Artificial Intelligence*, pages 228–233. 2011.

[104] A. Hevner and S. Chatterjee. Design research in information systems. *Integrated Series in Information Systems*, 22:pages 9–22, 2010.

[105] K. Hindriks and B. van Riemsdijk. Using temporal logic to integrate goals and qualitative preferences into agent programming. In *Proceedings of International Workshop on Declarative Agent Languages and Technologies*, pages 215–232. 2008.

[106] T. Hobbes. *Leviathan or The Matter, Forme and Power of a Common Wealth Ecclesiasticall and Civil*. Simon & Schuster, 1962.

[107] S. Holland. *Preference Mining and Preference Repositories: Design, Algorithms and Personalized Applications*. Ph.D. thesis, University of Augsburg, 2003.

[108] M. Hollis. *The Philosophy of Social Science: An Introduction*. Cambridge University Press, 1994.

[109] G. C. Homans. *The Human Group*. New York: Harcourt, 1950.

[110] J. F. Hübner, E. Matson, O. Boissier, and V. Dignum, editors. *Coordination, Organizations, Institutions, and Norms in Agent Systems IV*. 2009.

[111] J. F. Hübner, J. S. Sichman, and O. Boissier. Developing organised multi-agent systems using the MOISE+ model: programming issues at the system and agent levels. *International journal of agent-oriented software engineering*, 1:pages 370–395, 2007.

[112] K. Jensen. *Coloured Petri Nets: Basic Concepts, Analysis Methods and Practical Uses*. Springer, 1997.

[113] K. Jensen, L. M. Kristensen, and L. Wells. Coloured petri nets and cpn tools for modelling and validation of concurrent systems. *International Journal on Software Tools for Technology Transfer*, 9(3-4):pages 213–254, 2007.

[114] J. Jiang. OperA+: a model for context-aware organizational interactions in virtual organizations. In *Proceedings of the 12th International Conference on Autonomous Agents and MultiAgent Systems (AAMAS)*, pages 1435–1436. 2013.

[115] J. Jiang, H. Alderwereld, V. Dignum, and Y. H. Tan. Compliance checking of organizational interactions. *ACM Transactions on Management Information Systems*, 2015. To appear.

[116] J. Jiang, H. Alderwereld, V. Dignum, S. Z. Wang, and Z. Baida. Regulatory compliance of business processes. *AI and Society*, pages 1–10, 2014.

[117] J. Jiang, H. Aldewereld, V. Dignum, and Y. H. Tan. Norm contextualization. In *Coordination, Organisations, Institutions and Norms in Agent Systems VIII*, pages 141–157. 2013.

[118] J. Jiang, V. Dignum, H. Aldewereld, F. Dignum, and Y. H. Tan. Norm compliance checking. In *Proceeding of International Conference on Autonomous Agents and Multiagent Systems*, pages 1121–1122. 2013.

[119] J. Jiang, V. Dignum, and Y. H. Tan. An agent-based inter-organizational collaboration framework: OperA+. In *Proceeding of the 13th International Workshop on Coordination, Organisations, Institutions and Norms*, pages 58–74. 2011.

[120] J. Jiang, V. Dignum, Y. H. Tan, and S. Overbeek. A context-aware inter-organizational collaboration model applied to international trade. In *Proceeding of the International IFIP Electronic Government Conference*, pages 308–319. 2011.

[121] J. Jiang, B. Huisman, and V. Dignum. Agent-based multi-organizational interaction design: A case study of the dutch railway system. In *Proceedings of the 2012 IEEE/WIC/ACM International Conference on Intelligent Agent Technology (IAT)*, pages 196–203. 2012.

[122] J. Jiang, J. Thangarajah, H. Aldewereld, and V. Dignum. Reasoning with agent preferences in normative multi-agent systems. In *Proceeding of International Conference on Autonomous Agents and Multiagent Systems*, pages 1373–1374. 2014.

[123] A. J. I. Jones and M. Sergot. On the characterization of law and computer systems: the normative systems perspective. In *Deontic Logic in Computer Science: Normative System Specification*, pages 275–305. John Wiley & Sons Ltd., 1993.

[124] A. J. I. Jones and M. Sergot. A formal characterization of institutionalised power. *Journal of the IGPL*, 3:pages 427–443, 1996.

[125] S. Kaci, N. Patel, and V. Prince. From NL preference expressions to comparative preference statements: A preliminary study in eliciting preferences for customised decision support. In *Proceedings of the 26th International Conference on Tools with Artificial Intelligence*, pages 591–598. 2014.

[126] D. Katz and R. Kahn. *The social psychology of organizations*. New York: Wiley, 1966.

[127] P. Kaźmierczak, T. Pedersen, and T. Ågotnes. NORMC: a norm compliance temporal logic model checker. In *Proceedings of the 6th Starting AI Researchers' Symposium*, pages 168–179. 2012.

[128] M. E. Kharbili, A. K. A. de Medeiros, S. Stein, and W. M. P. van der Aalst. Business process compliance checking: Current state and future challenges. In *Modellierung Betrieblicher Informationssysteme*, pages 107–113. 2008.

[129] S. Kim. IT compliance of industrial information systems: Technology management and industrial engineering perspective. *Journal of Systems and Software*, 80(10):pages 1590–1593, 2007.

[130] M. Köhler. A formal model of multi-agent organisations. *Fundamenta Informaticae*, 79(3-4):pages 415 – 430, 2007.

[131] M. Köhler, D. Moldt, and H. Rölke. Modelling the structure and behaviour of petri net agents. In *Proceedings of ICATPN*, pages 224–241. 2001.

[132] M. Köhler-Bußmeier, D. Moldt, and M. Wester-Ebbinghaus. A formal model for organisational structures behind process-aware information systems. In *Transactions on Petri Nets and Other Models of Concurrency II: Special Issue on Concurrency in ProcessAware Information Systems*, pages 98–114. 2009.

[133] M. J. Kollingbaum, W. Vasconcelos, A. García-Camino, and T. J. Norman. Managing conflict resolution in norm-regulated environments. In *Engineering Societies in the Agents World VIII*, pages 55–71. 2008.

[134] O. Kummer. Introduction to petri nets and reference nets. *Sozionik Aktuell*, 1:pages 1–9, 2001.

[135] O. Kummer, F. Wienberg, and M. Duvigneau. Reference net workshop (renew). `http://www.renew.de`. Universitt Hamburg.

[136] M. de Leoni and W. M. P. van der Aalst. Aligning event logs and process models for multi-perspective conformance checking: An approach based on integer linear programming. In *Proceedings of the International conference on Business Process Management*, pages 113–129. 2013.

[137] M. de Leoni and W. M. P. van der Aalst. Data-aware process mining: Discovering decisions in processes using alignments. In *Proceedings of the 28th Annual ACM Symposium on Applied Computing*, pages 1454–1461. 2013.

[138] M. de Leoni, F. M. Maggi, and W. M. P. van der Aalst. Aligning Event Logs and Declarative Process Models for Conformance Checking. In *Proceedings of the International Conference on Business Process Management*, pages 82–97. 2012.

[139] Y. Lésperance, H. Levesque, F. Lin, D. Marcu, R. Reiter, and R. Scherl. Foundations of a logical approach to agent programming. In *Intelligent Agents II*, pages 331–346. 1996.

[140] T. Li, T. Balke, M. De Vos, K. Satoh, and J. Padget. Detecting conflicts in legal systems. In *New Frontiers in Artificial Intelligence*, pages 174–189. 2013.

[141] T. Li, J. Jiang, H. Aldewereld, M. De Vos, V. Dignum, and J. Padget. Contextualized institutions in virtual organizations. In *Coordination, Organisations, Institutions and Norms in Agent Systems VIIII*, pages 136–154. 2013.

[142] S. Lichtenstein and P. Slovic, editors. *The Construction of Preference*. Cambridge University Press, 2006.

[143] Y. Liu, S. Müller, and K. Xu. A static compliance-checking framework for business process models. *IBM System Journal*, 46(2):pages 335–361, 2007.

[144] N. Lohmann. Compliance by design for artifact-centric business processes. *Information Systems*, 38(4):pages 606–618, 2013.

[145] F. López y López and M. Luck. A model of normative multi-agent systems and dynamic relationships. In *Regulated Agent-Based Social Systems*, pages 259–280. 2004.

[146] F. López y López, M. Luck, and M. d'Inverno. A normative framework for agent-based systems. *Computational and Mathematical Organization Theory*, 12(2–3):pages 227–250, 2006.

[147] E. de Maat and R. Winkels. Automated classification of norms in sources of law. In *Semantic Processing of Legal Texts*, pages 170–191. 2010.

[148] P. Maes, editor. *Designing Autonomous Agents*. MIT Press, 1990.

[149] D. Makinson and L. van der Torre. Input/output logics. *J. Philosophical Logic*, 29(4):pages 383–408, 2000.

[150] T. F. Malloy. Regulation, compliance and the firm. *Temple Law Review*, 76:pages 1143–1158, 2003.

[151] C. Masolo, L. Vieu, E. Bottazzi, C. Catenacci, R. Ferrario, A. Gangemi, and N. Guarino. Social roles and their descriptions. In *Proceedings of KR*, pages 267–277. 2004.

[152] F. Meneguzzi, P. R. Telang, and M. P. Singh. A first-order formalization of commitments and goals for planning. In *Proceedings of AAAI*, pages 697–703. 2013.

[153] J.-J. Ch. Meyer. A different approach to deontic logic: Deontic logic viewed as a variant of dynamic logic. *Notre Dame Journal of Formal Logic*, 29(1):pages 109–136, 1988.

[154] J.-J. Ch. Meyer and R. Wieringa, editors. *Deontic Logic in Computer Science: Normative System Specification*. John Wiley & Sons Ltd., 1993.

[155] R. Milner, M. Tofte, R. Harper, and D. MacQueen. *The Definition of Standard ML*. MIT Press, 1997.

[156] R. B. Mitchell. Compliance theory: An overview. In J. Cameron, J. Werksman, and P. Roderick, editors, *Improving Compliance with International Environmental Law*, pages 3–28. 1996.

[157] M. Montali. Declarative process mining. In *Specification and Verification of Declarative Open Interaction Models*, pages 343–365. 2010.

[158] J. Müller, M. Pischel, and M. Thiel. Modelling reactive behavior in vertically layered agent architectures. In *Intelligent Agent: Theories, Architectures and Languages*, pages 261–276. 1995.

[159] T. Murata. Petri nets: Properties, analysis and applications. In *Proceedings of the IEEE*, pages 541–580. 1989.

[160] K. L. Myers and D. N. Morley. Resolving conflicts in agent guidance. In *Proceedings of the Workshop on Preferences in AI and CP: Symbolic Approaches*, pages 1–8. 2002.

[161] P. Noriega. *Agent mediated auctions: The Fishmarket Metaphor*. Ph.D. thesis, Universitat Autnoma de Barcelona, 1997.

[162] J. Odell, H. Van Dyke Parunak, and B. Bauer. Representing agent interaction protocols in UML. In *Proceeding of the first international workshop on Agent-oriented software engineering*, pages 121–140. 2001.

[163] D. E. O'Leary, D. Kuokka, and R. Plant. Artificial intelligence and virtual organizations. *Communications of the ACM*, 40(1):pages 52–59, 1997.

[164] A. Omicini. SODA: Societies and infrastructures in the analysis and design of agent-based systems. In *Agent-Oriented Software Engineering*, pages 185–193. 2001.

[165] E. Ostrom. *Understanding institutional diversity*. Princeton University Press, 2005.

[166] O. Pacheco and J. Carmo. A role based model for the normative specification of organized collective agency and agents interaction. *Autonomous Agents and Multi-Agent Systems*, 6(2):pages 145–184, 2003.

[167] T. Parsons. *Structure and process in modern societies*. Free Press, 1960.

[168] J. Pavón and J. Gómez-Sanz. Agent oriented software engineering with INGENIAS. In *Multi-Agent Systems and Applications III*, pages 394–403. 2003.

[169] C. A. Petri. *Kommunikation mit Automaten (Communication with automata)*. Ph.D. thesis, Bonn: Institut für Instrumentelle Mathematik, 1962.

[170] H. Prakken and M. J. Sergot. Contrary-to-duty obligations. *Studia Logica*, 57(1):pages 91–115, 1996.

[171] C. Prisacariu and G. Schneider. $\mathcal{CL}$: An action-based logic for reasoning about contracts. In *Proceedings of the 16th International Workshop on Logic, Language, Information and Computation*, pages 335–349. 2009.

[172] E. Ramezani, D. Fahland, and W. M. P. van der Aalst. Where did I misbehave? diagnostic information in compliance checking. In *Proceedings of the International Conference on Business Process Management*, pages 262–278. 2012.

[173] E. Ramezani, D. Fahland, and W. M. P. van der Aalst. Supporting domain experts to select and configure precise compliance rules. In *Proceedings of the Workshop on Security in Business Processes*, pages 498–512. 2014.

[174] E. Ramezani, D. Fahland, B. F. van Dongen, and W. M. P. van der Aalst. Diagnostic information in temporal compliance checking. Technical report, BPM Center, 2012.

[175] A. S. Rao and M. P. Georgeff. An abstract architecture for rational agents. In *Proceeding of International Conference on Principles of Knowledge Representation and Reasoning*, pages 439–449. 1992.

[176] J.-F. Raskin, B. Namur, L. W. N. van der Torre, and Y. H. Tan. How to model normative behavior in petri nets. In *Proceedings of the 2nd Modelage Workshop on Formal Models of Agents*, pages 223–241. 1996.

[177] J. Rawls. Two concepts of rules. *The Philosophical Review*, 64(1):pages 3–32, 1955.

[178] A. Ricci, M. Piunti, M. Viroli, and A. Omicini. Environment programming in CArtAgO. In *Multi-Agent Programming: Languages,Platforms and Applications*, pages 259–288. 2009.

[179] J. Rodriguez. *On the Design and Construction of Agent-mediated Electronic Institutions*. Ph.D. thesis, Inst. d'Investigació en Intel.ligència Artificial, 2001.

[180] A. Ross. *Directives and Norms*. New York: Humanities Press, 1968.

[181] N. Russell, A. H. M. ter Hofstede, W. M. P. van der Aalst, and N. Mulyar. Workflow control-flow patterns: A revised view. Technical Report BPM-06-22, BPM Center, 2006.

[182] S. Sadiq, G. Governatori, and K. Namiri. Modeling control objectives for business process compliance. In *Proceedings of the International Conference on Business Process Management*, pages 149–164. 2007.

[183] G. Sartor. Normative conflict in legal reasoning. *Artificial Intelligence and Law*, 1:pages 209–235, 1992.

[184] B. T. R. Savarimuthu, S. Cranefield, M. Purvis, and M. Purvis. A data mining approach to identify obligation norms in agent societies. In *Agents and Data Mining Interaction*, pages 43–58. 2010.

[185] C. Schneeweiss. Distributed decision making - a unified approach. *European Journal of Operational Research*, 150:pages 237–252, 2003.

[186] W. R. Scott. *Organizations: Rational, Natural and Open Systems*. Prentice Hall, 1992.

[187] J. R. Searle. *Speech Acts. An Essay in the Philosophy of Language*. TCambridge University Press, 1969.

[188] J. R. Searle. *The construction of social reality*. The free press, 1995.

[189] K. Segerberg. A deontic logic of action. *Studia Logica*, 41(2):pages 269–282, 1982.

[190] S. Sen and S. Airiau. Emergence of norms through social learning. In *Proceedings of the 20th International Joint Conference on Artifical Intelligence*, pages 1507–1512. 2007.

[191] Y. Shoham and M. Tennenholtz. On the emergence of social conventions: modeling, analysis, and simulations. *Artificial Intelligence*, 94(1–2):pages 139–166, 1997.

[192] H. A. Simon, editor. *Administrative Behavior (4th ed.)*. Simon and Schuster Inc., 1997.

[193] J. Simpson and E. Weiner, editors. *The Oxford English Dictionary*. Oxford University Press, 1989.

[194] M. P. Singh. An ontology for commitments in multiagent systems: Toward a unification of normative concepts. *Artificial Intelligence and Law*, 7(1):pages 97–113, 1999.

[195] M. P. Singh. Semantical considerations on dialectical and practical commitments. In *Proceedings of AAAI*, pages 176–181. 2008.

[196] M. P. Singh. Norms as a basis for governing sociotechnical systems. *ACM Transactions on Intelligent Systems and Technology*, 5(1):pages 1–23, 2013.

[197] T. C. Son and E. Pontelli. Planning with preferences using logic programming. *Theory and Practice of Logic Programming*, 6(5):pages 559–607, 2006.

[198] J. F. Sowa. *Conceptual Structures: Information Processing in Mind and Machine*. Addison-Wesley, New York, 1984.

[199] J. F. Sowa. Using a lexicon of canonical graphs in a semantic interpreter. In *Relational Models of the Lexicon: Representing Knowledge in Semantic Networks*, pages 113–137. 1988.

[200] F. Steimann. On the representation of roles in object-oriented and conceptual modelling. *Data & Knowledge Engineering*, 35:pages 83–106, 2000.

[201] Y. H. Tan, N. Bjørn-Andersen, S. Klein, and B. Rukanova, editors. *Accelerating Global Supply Chains with IT-Innovation*. Berlin: Springer-Verlag, 2011.

[202] Y. H. Tan and L. van der Torre. How to combine ordering and minimizing in a deontic logic based on preferences. In *Proceedings of DEON*, pages 216–232. 1996.

[203] N. A. M. Tinnemeier, M. Dastani, J.-J. Ch. Meyer, and L. van der Torre. Programming normative artifacts with declarative obligations and prohibitions. In *Proceedings of IAT*, pages 145–152. 2009.

[204] S. C. Tosatto, G. Governatori, and P. Kelsen. Detecting deontic conflicts in dynamic settings. In *Deontic Logic and Normative Systems*, pages 65–80. 2014.

[205] T. R. Tyler and S. L. Blader. Can businesses effectively regulate employee conduct? the antecents of rule following in work settings. *The Academy of Management Journal*, 48(6):pages 1143–1158, 2005.

[206] W. Vasconcelos, M. Kollingbaum, and T. Norman. Normative conflict resolution in multiagent systems. *Autonomous Agents and Multi-Agent Systems*, 19(2):pages 124–152, 2009.

[207] W. Vasconcelos, M. J. Kollingbaum, and T. J. Norman. Resolving conflict and inconsistency in norm-regulated virtual organizations. In *Proceedings of the 6th International Joint Conference on Autonomous Agents and Multiagent Systems*, pages 644–651. 2007.

[208] J. Vázquez-Salceda. *The role of norms and electronic institutions in multi-agent systems applied to complex domains. The HARMONIA framework*. Birkhäuser, 2003.

[209] M. Venkatraman and M. P. Singh. Verifying compliance with commitment protocols. *Autonomous Agents and Multi-Agent Systems*, 2(3):pages 217–236, 1999.

[210] S. Visser, J. Thangarajah, and J. Harland. Reasoning about preferences in intelligent agent systems. In *Proceedings of the International Joint Conference on Artificial Intelligence*, pages 426–431. 2011.

[211] L. Volonino, G. H. Gessner, and G. F. Kermis. Holistic compliance with sarbanes-oxley. *Communications of the Association for Information Systems*, 14(1):pages 219–233, 2004.

[212] M. Weber. *From Max Weber: Essays in sociology*. Oxford University Press, 1946.

[213] M. Weber. *The theory of social and economic organization*. Free Press, 1946.

[214] M. Wenzel. Motivation or rationalisation? causal relations between ethics, norms and tax compliance. *Journal of Economic Psychology*, 26(4):pages 491–508, 2005.

[215] M. Wooldridge. Intelligent agents. In G. Weiss, editor, *Multiagent Systems: A Modern Approach to Distributed Modern Approach to Artificial Intelligence*, pages 27–78. 1999.

[216] M. Wooldridge. *An Introduction to MultiAgent Systems*. John Wiley & Sons, 2009.

[217] M. Wooldridge and N. R. Jennings. Intelligent agents: Theory and practice. *Knowledge engineering review*, 10(2):pages 115–152, 1995.

[218] G. H. von Wright. Deontic logic. *Mind*, 60:pages 1–15, 1951.

[219] G. H. von Wright. A new system of deontic logic. *Danish Yearbook of Philisophy*, pages 173–182, 1983.

[220] G. H. von Wright. Norms, truth and logic. *In Practical Reason*, I:pages 130–209, 1983.

[221] P. Yolum and M. P. Singh. Reasoning about commitments in the event calculus: An approach for specifying and executing protocols. *Ann. Math. Artif. Intell.*, 42(1-3):pages 227–253, 2004.

[222] D. Zaelke, D. Kaniaru, and E. Kružíková, editors. *Making Law Work: Environmental Compliance & Sustainable Development*. International Law Publishers, 2005.

[223] F. Zambonelli, N. R. Jennings, and M. Wooldridge. Multi-agent systems as computational organizations: The Gaia methodology. In *Agent-oriented Methodologies*, pages 136–171. 2005.

[224] A. Zimmermann, A. Lorenz, and R. Oppermann. An operational definition of context. In *Modeling and Using Context*, pages 558–571. 2007.

Summary

The motivation of this research comes from the need of devising and evaluating solutions to achieving organizational compliance, which is an important factor in ensuring the success of business operations, the safety of business environments and the social welfare. With the dramatic increase in both the amount and complexity of compliance requirements, organizations are confronted with challenges in handling the dynamics of organizational compliance with respect to changing environments/contexts, dealing with the combination of multiple sources of compliance requirements and facilitating the reuse of solutions across different contexts. Moreover, since organizational participants are autonomous entities that are capable of controlling their own actions to pursue personal satisfaction, it is necessary to take into account individual preferences in both the design and evaluation of solutions to achieving organizational compliance.

The OperA+ framework presented in this dissertation provides a set of formalisms and mechanisms for the representation and evaluation of organizational compliance. It uses organizational theory and normative multi-agent systems as the basis of conceptual design. An organization is specified in OperA+ by three interrelated structures, i.e., Social Structure, Normative Structure and Preference Structure. The Social Structure, based on role theory, specifies how compliance requirements are encapsulated through the identification of roles which present the standards for the recruitment of agents in an organization and link the evaluation to the agents' performance. By introducing the concept of composite roles, we are able to design and analyze organizations at different scales from abstract values to implementation details. The Normative Structure, based on Searle's notion of the construction of social reality and deontic logic, specifies the compliance rules of organizational control by means of constitutive norms and regulative norms. Taking into account the interrelations between norms, we devise a method, based on modularization, of formalizing normative structures from sets of compliance rules. The combination of the Social Structure and Normative Structure provides a representation of organizational compliance by indicating what desired behavior (norms) should be ensured by whom (role enacting agents). The Preference Structure, based on preference languages in the domain of automated planning, specifies the individual preferences of the participating agents in organizations, providing a formalization of individual autonomy.

In OperA+, environmental characteristics of organizational compliance are captured by the notion of contexts, based on the observation that different organizational behaviors may be expected when the environment in which the organization exists changes. Combined with the compositional role model in the social structure, OperA+ enables a multi-level and multi-context modeling paradigm. This not only enables designers to characterize the situations to which specific compliance requirements are applied, but also adds meaningful links between different levels of descriptions and facilitates the modularization of organizational design into reusable blocks. Such characteristics of OperA+ enhance organizational compliance modeling and reasoning in changing environments.

In OperA+, the evaluation of organizational compliance consists of two parts. The first part relates to the normative structure, in which both conflict detection and compliance checking of norms are considered. Conflict detection determines whether there are any norms that cannot be complied with simultaneously, providing a mechanism for the analysis of compliance requirements from different regulation sources. Compliance checking determines whether the participants' behavior violates any norms (obligations and prohibitions) specified in the normative structure. The second part relates to the preference structure, in which we incorporate agent autonomy into the evaluation of organizational compliance by including the analysis of preference satisfaction. The evaluation of organizational compliance is operationalized via a computational model developed on the basis of Colored Petri Nets (CPNs). By leveraging the existing analytical tools of CPNs, we unify the analysis of norm compliance and preference satisfaction, which facilitates the process of finding a balance between comformity and autonomy in achieving organizational compliance.

We demonstrate the applicability of OperA+ by conducting a series of case studies concerning organizational interactions in railway systems and international trade systems. The case studies provide evidence on how organizational compliance can be modeled and analyzed by OperA+. We summary the case studies by a list of guidelines in using OperA+. The framework is the first step towards an integral approach to the most important aspects in the design and evaluation of organizational compliance.

Samenvatting

De motivatie van dit onderzoek komt vanuit de noodzaak om oplossingen voor het bereiken van organizationele conformiteit te creren en te evalueren. Conformiteit is een belangrijke succesfactor voor de bedrijfsvoering, de veiligheid in de bedrijfsomgeving en het maatschappelijke welzijn. De enorme toename in zowel de hoeveelheid als de complexiteit van conformiteitseisen biedt een uitdaging voor organizaties om om te gaan met de dynamiek van organizationele conformiteit ten opzichte van veranderende omgevingen/contexten, met verschillende bronnen van conformiteitseisen, en het faciliteren van het hergebruik van oplossingen binnen verschillende contexten. Verder, omdat organizationele participanten autonome entiteiten zijn, die in staat zijn om hun eigen acties te bepalen en hun eigen doelen nastreven, is het noodzakelijk om individuele voorkeuren te betrekken in het ontwerp en de evaluatie van de oplossingen die organizationele conformiteit bereiken.

De OperA+ framework, gepresenteerd in dit proefschrift, biedt formalismen en mechanieken voor het representeren en evalueren van organizationele conformiteit. Het maakt gebruik van organizatietheorie en normatieve agentsystemen als de basis voor het conceptuele ontwerp. Een organizatie wordt in OperA+ gerepresenteerd door drie onderling verbonden structuren, namelijk: Social Structure, Normative Structure, en Preference Structure. De Social Structure, gebaseerd op role theory, specificeert hoe de conformiteitseisen zijn ingebed in de organizatie door de identificatie van de rollen die de standaarden zijn voor het recruiten van agents in de organizatie en die de verbinding vormt naar de evaluatie van de prestaties van de agents. Door de introductie van samengestelde rollen zijn we in staat om het ontwerp en de analyses van de organizatie te maken op een schaal van de abstracte waarden tot de implementatie details. De Normative Structure, gebaseerd op Searles conceptualisatie van de constructie van sociale realiteit en deontische logica, specificeert de conformiteitsregels van organizationele controle door het gebruik van constitutieve en regulatieve normen. Met het in oog houden van de onderlinge relaties tussen normen, hebben wij een methode ontwikkeld, gebaseerd op modularisatie, om normatieve structuren te formuleren vanuit verzamelingen van conformiteitsregels. De combinatie van de Social Structure en de Normative Structure biedt een representatie van organizationele conformiteit door aan te geven welk het verwacht gedrag (de normen) moet worden bereikt door wie (agents die een rol spelen). De Preference Structure, gebaseerd op preference languages uit het gebied van automatische planning, specificeert de individuele voorkeuren van de deelnemende agents in de organizaties, en biedt daarmee een formalisatie van de individuele autonomie.

In OperA+ worden de omgevingseigenschappen van organizationele conformiteit omvat door de noties van context, gebaseerd op de observatie dat verschillende organizationele gedragingen verwacht kunnen worden als de omgeving waarin de organizatie gesitueerd is verandert. Gecombineerd met het samengestelde rollenmodel in de Social Structure, maakt OperA+ een modelleerparadigma voor meerdere niveaus en contexten. Naast dat dit ontwerpers in staat stelt om de situaties van organizationele conformiteit te karakteriseren, zorgt het er ook voor dat betekenisvolle relaties gelegd kunnen worden tussen verschillende niveaus van beschrijving en vergemakkelijkt het daarmee de modularisatie van organizationeel ontwerp door de introductie van herbruikbare blokken. Deze eigenschappen van OperA+ versterken het modelleren van organizationele conformiteit en het redeneren in veranderende omgevingen.

In OperA+ bestaat de evaluatie van organizationele conformiteit uit twee onderdelen. Het eerste onderdeel is gerelateerd aan de Normative Structure, waarin zowel conflict detectie als conformiteitschecking worden beschouwd. De conflict detectie bepaalt of er normen zijn waaraan niet kan worden voldaan op hetzelfde moment. Dit biedt een mechanisme voor het analyseren van de conformiteitseisen uit meerdere regulatieve bronnen. De conformiteitscheck bepaalt of een van de participanten normen (obligaties of verboden) van de Normative Structure overtreden. Het tweede onderdeel is gerelateerd aan de Preference Structure, waarin we de autonomie van de agents gebruiken in de evaluatie van de organizationele conformiteit door de analyse van de volbrachte van individuele voorkeuren. De evaluatie van organizationele conformiteit is geoperationaliseerd door de ontwikkeling van een computationeel model gebaseerd op Colored Petri Nets (CPNs). Door gebruik te maken van de bestaande analytische mogelijkheden van CPNs, hebben we een uniforme analyse gemaakt voor de conflict detectie en conformiteitscheck, waardoor het vinden van een balans tussen de conformiteit en autonomie makkelijker wordt gemaakt bij het trachten te bereiken van organizationele conformiteit.

We hebben de toepasbaarheid van OperA+ gedemonstreerd door het doen van een reeks van casestudies naar organizationele interactie in treinsystemen en internationale handelssystemen. De casestudies bewijzen hoe organizationele conformiteit gemodelleerd en geanalyseerd kan worden in OperA+. Uit de casestudies hebben we een lijst met richtlijnen voor het gebruik van OperA+ samengevat. Het

OperA+ framework is de eerste stap naar een integrale aanpak voor de belangrijkste stappen in het ontwerp en evaluatie van organizationele conformiteit.

Curriculum-vitae

Jie Jiang was born in Xi'an, China on November 10, 1985. Upon completing high school education in 2003, she started studying control theory and control engineering at Chang'an University in China. She obtained her bachelor's degree in 2007, and continued to study towards a master's degree at Xi'an Jiaotong University in the research area of workflow management systems. To carry on her interest in how computer technologies can facilitate management, Jie started her PhD research at Delft University of Technology in September 2010, with a special focus on the analysis and verification of organizational behavior. In her research, she explored how organizational compliance can be ensured through organizational design, taking into account various regulation sources, the environmental characteristics, and the autonomy of organizational participants. Combining Normative Multi-agent Systems, Context-enabled Organization Modeling and Colored Petri Nets, she devised a framework consisting of formalisms and mechanisms for modeling and evaluating organizational compliance.

SIKS Dissertation Series

2009

2009-01: **Rasa Jurgelenaite** (RUN), *Symmetric Causal Independence Models*

2009-02: **Willem Robert van Hage** (VU), *Evaluating Ontology-Alignment Techniques*

2009-03: **Hans Stol** (UvT), *A Framework for Evidence-based Policy Making Using IT*

2009-04: **Josephine Nabukenya** (RUN), *Improving the Quality of Organisational Policy Making using Collaboration Engineering*

2009-05: **Sietse Overbeek** (RUN), *Bridging Supply and Demand for Knowledge Intensive Tasks - Based on Knowledge, Cognition, and Quality*

2009-06: **Muhammad Subianto** (UU), *Understanding Classification*

2009-07: **Ronald Poppe** (UT), *Discriminative Vision-Based Recovery and Recognition of Human Motion*

2009-08: **Volker Nannen** (VU), *Evolutionary Agent-Based Policy Analysis in Dynamic Environments*

2009-09: **Benjamin Kanagwa** (RUN), *Design, Discovery and Construction of Service-oriented Systems*

2009-10: **Jan Wielemaker** (UVA), *Logic programming for knowledge-intensive interactive applications*

2009-11: **Alexander Boer** (UVA), *Legal Theory, Sources of Law & the Semantic Web*

2009-12: **Peter Massuthe** (TUE, Humboldt-Universitaet zu Berlin), *perating Guidelines for Services*

2009-13: **Steven de Jong** (UM), *Fairness in Multi-Agent Systems*

2009-14: **Maksym Korotkiy** (VU), *From ontology-enabled services to service-enabled ontologies (making ontologies work in e-science with ONTO-SOA)*

2009-15: **Rinke Hoekstra** (UVA), *Ontology Representation - Design Patterns and Ontologies that Make Sense*

2009-16: **Fritz Reul** (UvT), *New Architectures in Computer Chess*

2009-17: **Laurens van der Maaten** (UvT), *Feature Extraction from Visual Data*

2009-18: **Fabian Groffen** (CWI), *Armada, An Evolving Database System*

2009-19: **Valentin Robu** (CWI), *Modeling Preferences, Strategic Reasoning and Collaboration in Agent-Mediated Electronic Markets*

2009-20: **Bob van der Vecht** (UU), *Adjustable Autonomy: Controling Influences on Decision Making*

2009-21: **Stijn Vanderlooy** (UM), *Ranking and Reliable Classification*

2009-22: **Pavel Serdyukov** (UT), *Search For Expertise: Going beyond direct evidence*

2009-23: **Peter Hofgesang** (VU), *Modelling Web Usage in a Changing Environment*

2009-24: **Annerieke Heuvelink** (VUA), *Cognitive Models for Training Simulations*

2009-25: **Alex van Ballegooij** (CWI), *"RAM: Array Database Management through Relational Mapping"*

2009-26: **Fernando Koch** (UU), *An Agent-Based Model for the Development of Intelligent Mobile Services*

2009-27: **Christian Glahn** (OU), *Contextual Support of social Engagement and Reflection on the Web*

2009-28: **Sander Evers** (UT), *Sensor Data Management with Probabilistic Models*

2009-29: **Stanislav Pokraev** (UT), *Model-Driven Semantic Integration of Service-Oriented Applications*

2009-30: **Marcin Zukowski** (CWI), *Balancing vectorized query execution with bandwidth-optimized storage*

2009-31: **Sofiya Katrenko** (UVA), *A Closer Look at Learning Relations from Text*

2009-32: **Rik Farenhorst (VU) and Remco de Boer** (VU), *Architectural Knowledge Management: Supporting Architects and Auditors*

2009-33: **Khiet Truong** (UT), *How Does Real Affect Affect Affect Recognition In Speech?*

2009-34: **Inge van de Weerd** (UU), *Advancing in Software Product Management: An Incremental Method Engineering Approach*

2009-35: **Wouter Koelewijn** (UL), *Privacy en Politiegegevens; Over geautomatiseerde normatieve informatie-uitwisseling*

2009-36: **Marco Kalz** (OUN), *Placement Support for Learners in Learning Networks*

2009-37: **Hendrik Drachsler** (OUN), *Navigation Support for Learners in Informal Learning Networks*

2009-38: **Riina Vuorikari** (OU), *Tags and self-organisation: a metadata ecology for learning resources in a multilingual context*

2009-39: **Christian Stahl** (TUE, Humboldt-Universitaet zu Berlin), *Service Substitution – A Behavioral Approach Based on Petri Nets*

2009-40: **Stephan Raaijmakers** (UvT), *Multinomial Language Learning: Investigations into the Geometry of Language*

2009-41: **Igor Berezhnyy** (UvT), *Digital Analysis of Paintings*

2009-42: Toine Bogers *Recommender Systems for Social Bookmarking*

2009-43: **Virginia Nunes Leal Franqueira** (UT), *Finding Multi-step Attacks in Computer Networks using Heuristic Search and Mobile Ambients*

2009-44: **Roberto Santana Tapia** (UT), *Assessing Business-IT Alignment in Networked Organizations*

2009-45: **Jilles Vreeken** (UU), *Making Pattern Mining Useful*

2009-46: **Loredana Afanasiev** (UvA), *Querying XML: Benchmarks and Recursion*

2010

2010-01: **Matthijs van Leeuwen** (UU), *Patterns that Matter*

2010-02: **Ingo Wassink** (UT), *Work flows in Life Science*

2010-03: **Joost Geurts** (CWI), *A Document Engineering Model and Processing Framework for Multimedia documents*

2010-04: **Olga Kulyk** (UT), *Do You Know What I Know? Situational Awareness of Co-located Teams in Multidisplay Environments*

2010-05: **Claudia Hauff** (UT), *Predicting the Effectiveness of Queries and Retrieval Systems*

2010-06: **Sander Bakkes** (UvT), *Rapid Adaptation of Video Game AI*

2010-07: **Wim Fikkert** (UT), *Gesture interaction at a Distance*

2010-08: **Krzysztof Siewicz** (UL), *Towards an Improved Regulatory Framework of Free Software. Protecting user freedoms in a world of software communities and eGovernments*

2010-09: **Hugo Kielman** (UL), *A Politiele gegevensverwerking en Privacy, Naar een effectieve waarborging*

2010-10: **Rebecca Ong** (UL), *Mobile Communication and Protection of Children*

2010-11: **Adriaan Ter Mors** (TUD), *The world according to MARP: Multi-Agent Route Planning*

2010-12: **Susan van den Braak** (UU), *Sensemaking software for crime analysis*

2010-13: **Gianluigi Folino** (RUN), *High Performance Data Mining using Bio-inspired techniques*

2010-14: **Sander van Splunter** (VU), *Automated Web Service Reconfiguration*

2010-15: **Lianne Bodenstaff** (UT), *Managing Dependency Relations in Inter-Organizational Models*

2010-16: **Sicco Verwer** (TUD), *Efficient Identification of Timed Automata, theory and practice*

2010-17: **Spyros Kotoulas** (VU), *Scalable Discovery of Networked Resources: Algorithms, Infrastructure, Applications*

2010-18: **Charlotte Gerritsen** (VU), *Caught in the Act: Investigating Crime by Agent-Based Simulation*

2010-19: **Henriette Cramer** (UvA), *People's Responses to Autonomous and Adaptive Systems*

2010-20: **Ivo Swartjes** (UT), *Whose Story Is It Anyway? How Improv Informs Agency and Authorship of Emergent Narrative*

2010-21: **Harold van Heerde** (UT), *Privacy-aware data management by means of data degradation*

2010-22: **Michiel Hildebrand** (CWI), *End-user Support for Access to Heterogeneous Linked Data*

2010-23: **Bas Steunebrink** (UU), *The Logical Structure of Emotions*

2010-24: **Dmytro Tykhonov** *Designing Generic and Efficient Negotiation Strategies*

2010-25: **Zulfiqar Ali Memon** (VU), *Modelling Human-Awareness for Ambient Agents: A Human Mindreading Perspective*

2010-26: **Ying Zhang** (CWI), *XRPC: Efficient Distributed Query Processing on Heterogeneous XQuery Engines*

2010-27: **Marten Voulon** (UL), *Automatisch contracteren*

2010-28: **Arne Koopman** (UU), *Characteristic Relational Patterns*

2010-29: **Stratos Idreos** (CWI), *Database Cracking: Towards Auto-tuning Database Kernels*

2010-30: **Marieke van Erp** (UvT), *Accessing Natural History - Discoveries in data cleaning, structuring, and retrieval*

2010-31: **Victor de Boer** (UVA), *Ontology Enrichment from Heterogeneous Sources on the Web*

2010-32: **Marcel Hiel** (UvT), *An Adaptive Service Oriented Architecture: Automatically solving Interoperability Problems*

2010-33: **Robin Aly** (UT), *Modeling Representation Uncertainty in Concept-Based Multimedia Retrieval*

2010-34: **Teduh Dirgahayu** (UT), *Interaction Design in Service Compositions*

2010-35: **Dolf Trieschnigg** (UT), *Proof of Concept: Concept-based Biomedical Information Retrieval*

2010-36: **Jose Janssen** (OU), *Paving the Way for Lifelong Learning; Facilitating competence development through a learning path specification*

2010-37: **Niels Lohmann** (TUE), *Correctness of services and their composition*

2010-38: **Dirk Fahland** (TUE), *From Scenarios to components*

2010-39: **Ghazanfar Farooq Siddiqui** (VU), *Integrative modeling of emotions in virtual agents*

2010-40: **Mark van Assem** (VU), *Converting and Integrating Vocabularies for the Semantic Web*

2010-41: **Guillaume Chaslot** (UM), *Monte-Carlo Tree Search*

2010-42: **Sybren de Kinderen** (VU), *Needs-driven service bundling in a multi-supplier setting – the computational e3-service approach*

2010-43: **Peter van Kranenburg** (UU), *A Computational Approach to Content-Based Retrieval of Folk Song Melodies*

2010-44: **Pieter Bellekens** (TUE), *An Approach towards Context-sensitive and User-adapted Access to Heterogeneous Data Sources, Illustrated in the Television Domain*

2010-45: **Vasilios Andrikopoulos** (UvT), *A theory and model for the evolution of software services*

2010-46: **Vincent Pijpers** (VU), *e3alignment: Exploring Inter-Organizational Business-ICT Alignment*

2010-47: **Chen Li** (UT), *Mining Process Model Variants: Challenges, Techniques, Examples*

2010-48: **Milan Lovric** (EUR), *Behavioral Finance and Agent-Based Artificial Markets*

2010-49: **Jahn-Takeshi Saito** (UM), *Solving difficult game positions*

2010-50: **Bouke Huurnink** (UVA), *Search in Audiovisual Broadcast Archives*

2010-51: **Alia Khairia Amin** (CWI), *Understanding and supporting information seeking tasks in multiple sources*

2010-52: **Peter-Paul van Maanen** (VU), *Adaptive Support for Human-Computer Teams: Exploring the Use of Cognitive Models of Trust and Attention*

2010-53: **Edgar Meij** (UVA), *Combining Concepts and Language Models for Information Access*

2011

2011-01: **Botond Cseke** (RUN), *Variational Algorithms for Bayesian Inference in Latent Gaussian Models*

2011-02: **Nick Tinnemeier** (UU), *Organizing Agent Organizations. Syntax and Operational Semantics of an Organization-Oriented Programming Language*

2011-03: **Jan Martijn van der Werf** (TUE), *Compositional Design and Verification of Component-Based Information Systems*

2011-04: **Hado van Hasselt** (UU), *Insights in Reinforcement Learning; Formal analysis and empirical evaluation of temporal-difference learning algorithms*

2011-05: **Base van der Raadt** (VU), *Enterprise Architecture Coming of Age - Increasing the Performance of an Emerging Discipline*

2011-06: **Yiwen Wang** (TUE), *Semantically-Enhanced Recommendations in Cultural Heritage*

2011-07: **Yujia Cao** (UT), *Multimodal Information Presentation for High Load Human Computer Interaction*

2011-08: **Nieske Vergunst** (UU), *BDI-based Generation of Robust Task-Oriented Dialogues* 2011-09: **Tim de Jong** (OU), *Contextualised Mobile Media for Learning*

2011-10: **Bart Bogaert** (UvT), *Cloud Content Contention*

2011-11: **Dhaval Vyas** (UT), *Designing for Awareness: An Experience-focused HCI Perspective*

2011-12: **Carmen Bratosin** (TUE), *Grid Architecture for Distributed Process Mining*

2011-13: **Xiaoyu Mao** (UvT), *Airport under Control. Multiagent Scheduling for Airport Ground Handling*

2011-14: **Milan Lovric** (EUR), *Behavioral Finance and Agent-Based Artificial Markets*

2011-15: **Marijn Koolen** (UvA), *The Meaning of Structure: the Value of Link Evidence for Information Retrieval*

2011-16: **Maarten Schadd** (UM), *Selective Search in Games of Different Complexity*

2011-17: **Jiyin He** (UVA), *Exploring Topic Structure: Coherence, Diversity and Relatedness*

2011-18: **Mark Ponsen** (UM), *Strategic Decision-Making in complex games*

2011-19: **Ellen Rusman** (OU), *The Mind ' s Eye on Personal Profiles*

2011-20: **Qing Gu** (VU), *Guiding service-oriented software engineering - A view-based approach*

2011-21: **Linda Terlouw** (TUD), *Modularization and Specification of Service-Oriented Systems*

2011-22: **Junte Zhang** (UVA), *System Evaluation of Archival Description and Access*

2011-23: **Wouter Weerkamp** (UVA), *Finding People and their Utterances in Social Media*

2011-24: **Herwin van Welbergen** (UT), *Behavior Generation for Interpersonal Coordination with Virtual Humans On Specifying, Scheduling and Realizing Multimodal Virtual Human Behavior*

2011-25: **Syed Waqar ul Qounain Jaffry** (VU), *Analysis and Validation of Models for Trust Dynamics*

2011-26: **Matthijs Aart Pontier** (VU), *Virtual Agents for Human Communication - Emotion Regulation and Involvement-Distance Trade-Offs in Embodied Conversational Agents and Robots*

2011-27: **Aniel Bhulai** (VU), *Dynamic website optimization through autonomous management of design patterns*

2011-28: Rianne Kaptein(UVA) *Effective Focused Retrieval by Exploiting Query Context and Document Structure*

2011-29: **Faisal Kamiran** (TUE), *Discrimination-aware Classification*

2011-30: **Egon van den Broek** (UT), ***Affective Signal Processing* (ASP),: Unraveling the mystery of emotions**

2011-31: **Ludo Waltman** (EUR), *Computational and Game-Theoretic Approaches for Modeling Bounded Rationality*

2011-32: **Nees-Jan van Eck** (EUR), *Methodological Advances in Bibliometric Mapping of Science*

2011-33: **Tom van der Weide** (UU), *Arguing to Motivate Decisions*

2011-34: **Paolo Turrini** (UU), *Strategic Reasoning in Interdependence: Logical and Game-theoretical Investigations*

2011-35: **Maaike Harbers** (UU), *Explaining Agent Behavior in Virtual Training*

2011-36: **Erik van der Spek** (UU), *Experiments in serious game design: a cognitive approach*

2011-37: **Adriana Burlutiu** (RUN), *Machine Learning for Pairwise Data, Applications for Preference Learning and Supervised Network Inference*

2011-38: **Nyree Lemmens** (UM), *Bee-inspired Distributed Optimization*

2011-39: **Joost Westra** (UU), *Organizing Adaptation using Agents in Serious Games*

2011-40: **Viktor Clerc** (VU), *Architectural Knowledge Management in Global Software Development*

2011-41: **Luan Ibraimi** (UT), *Cryptographically Enforced Distributed Data Access Control*

2011-42: **Michal Sindlar** (UU), *Explaining Behavior through Mental State Attribution*

2011-43: **Henk van der Schuur** (UU), *Process Improvement through Software Operation Knowledge*

2011-44: **Boris Reuderink** (UT), *Robust Brain-Computer Interfaces*

2011-45: **Herman Stehouwer** (UvT), *Statistical Language Models for Alternative Sequence Selection*

2011-46: **Beibei Hu** (TUD), *Towards Contextualized Information Delivery: A Rule-based Architecture for the Domain of Mobile Police Work*

2011-47: Azizi Bin Ab Aziz(VU) *Exploring Computational Models for Intelligent Support of Persons with Depression*

2011-48: **Mark Ter Maat** (UT), *Response Selection and Turn-taking for a Sensitive Artificial Listening Agent*

2011-49: **Andreea Niculescu** (UT), *Conversational interfaces for task-oriented spoken dialogues: design aspects influencing interaction quality*

2012

2012-01: **Terry Kakeeto** (UvT), *Relationship Marketing for SMEs in Uganda*

2012-02: **Muhammad Umair** (VU), *Adaptivity, emotion, and Rationality in Human and Ambient Agent Models*

2012-03: **Adam Vanya** (VU), *Supporting Architecture Evolution by Mining Software Repositories*

2012-04: **Jurriaan Souer** (UU), *Development of Content Management System-based Web Applications*

2012-05: **Marijn Plomp** (UU), *Maturing Interorganisational Information Systems*

2012-06: **Wolfgang Reinhardt** (OU), *Awareness Support for Knowledge Workers in Research Networks*

2012-07: **Rianne van Lambalgen** (VU), *When the Going Gets Tough: Exploring Agent-based Models of Human Performance under Demanding Conditions*

2012-08: **Gerben de Vries** (UVA), *Kernel Methods for Vessel Trajectories*

2012-09: **Ricardo Neisse** (UT), *Trust and Privacy Management Support for Context-Aware Service Platforms*

2012-10: **David Smits** (TUE), *Towards a Generic Distributed Adaptive Hypermedia Environment*

2012-11: **J.C.B. Rantham Prabhakara** (TUE), *Process Mining in the Large: Preprocessing, Discovery, and Diagnostics*

2012-12: **Kees van der Sluijs** (TUE), *Model Driven Design and Data Integration in Semantic Web Information Systems*

2012-13: **Suleman Shahid** (UvT), *Fun and Face: Exploring non-verbal expressions of emotion during playful interactions*

2012-14 **Evgeny Knutov** (TUE), *Generic Adaptation Framework for Unifying Adaptive Web-based Systems*

2012-15 **Natalie van der Wal** (VU), *Social Agents. Agent-Based Modelling of Integrated Internal and Social Dynamics of Cognitive and Affective Processes*

2012-16: **Fiemke Both** (VU), *Helping people by understanding them – Ambient Agents supporting task execution and depression treatment*

2012-17: **Amal Elgammal** (UvT), *Towards a Comprehensive Framework for Business Process Compliance*

2012-18: **Eltjo Poort** (VU), *Improving Solution Architecting Practices*

2012-19 **Helen Schonenberg** (TUE), *What's Next? Operational Support for Business Process Execution*

2012-20: **Ali Bahramisharif** (RUN), *Covert Visual Spatial Attention, a Robust Paradigm for Brain-Computer Interfacing*

2012-21 **Roberto Cornacchia** (TUD), *Querying Sparse Matrices for Information Retrieval*

2012-22: **Thijs Vis** (UvT), *Intelligence, politie en veiligheidsdienst: verenigbare grootheden?*

2012-23: **Christian Muehl** (UT), *Toward Affective Brain-Computer Interfaces: Exploring the Neurophysiology of Affect during Human Media Interaction*

2012-24: **Laurens van der Werff** (UT), *Evaluation of Noisy Transcripts for Spoken Document Retrieval*

2012-25: **Silja Eckartz** (UT), *Managing the Business Case Development in Inter-Organizational IT Projects: A Methodology and its Application*

2012-26: **Emile de Maat** (UVA), *Making Sense of Legal Text*

2012-27: **Hayrettin Gurkok** (UT), *Mind the Sheep! User Experience Evaluation & Brain-Computer Interface Games*

2012-28: **Nancy Pascall** (UvT), *Engendering Technology Empowering Women*

2012-29: **Almer Tigelaar** (UT), *Peer-to-Peer Information Retrieval*

2012-30: **Alina Pommeranz** (TUD), *Designing Human-Centered Systems for Reflective Decision Making*

2012-31: **Emily Bagarukayo** (RUN), *A Learning by Construction Approach for Higher Order Cognitive Skills Improvement, Building Capacity and Infrastructure*

2012-32: **Wietske Visser** (TUD), *Qualitative multi-criteria preference representation and reasoning*

2012-33: **Rory Sie** (OUN), **Coalitions in Cooperation Networks** (COCOON),

2012-34: **Pavol Jancura** (RUN), *Evolutionary analysis in PPI networks and applications*

2012-35: **Evert Haasdijk** (VU), *Never Too Old To Learn – On-line Evolution of Controllers in Swarm- and Modular Robotics*

2012-36: **Denis Ssebugwawo** (RUN), *Analysis and Evaluation of Collaborative Modeling Processes*

2012-37: **Agnes Nakakawa** (RUN), *A Collaboration Process for Enterprise Architecture Creation*

2012-38: **Selmar Smit** (VU), *Parameter Tuning and Scientific Testing in Evolutionary Algorithms*

2012-39: **Hassan Fatemi** (UT), *Risk-aware design of value and coordination networks*

2012-40: **Agus Gunawan** (UvT), *Information Access for SMEs in Indonesia*

2012-41: **Sebastian Kelle** (OU), *Game Design Patterns for Learning*

2012-42: **Dominique Verpoorten** (OU), *Reflection Amplifiers in self-regulated Learning*

2012-43: *Withdrawn*

2012-44: **Anna Tordai** (VU), *On Combining Alignment Techniques*

2012-45: **Benedikt Kratz** (UvT), *A Model and Language for Business-aware Transactions*

2012-46: **Simon Carter** (UVA), *Exploration and Exploitation of Multilingual Data for Statistical Machine Translation*

2012-47: **Manos Tsagkias** (UVA), *Mining Social Media: Tracking Content and Predicting Behavior*

2012-48: **Jorn Bakker** (TUE), *Handling Abrupt Changes in Evolving Time-series Data*

2012-49: **Michael Kaisers** (UM), *Learning against Learning – Evolutionary dynamics of reinforcement learning algorithms in strategic interactions*

2012-50: **Steven van Kervel** (TUD), *Ontologogy driven Enterprise Information Systems Engineering*

2012-51: **Jeroen de Jong** (TUD), *Heuristics in Dynamic Sceduling; a practical framework with a case study in elevator dispatching*

2013

2013-01: **Viorel Milea** (EUR), *News Analytics for Financial Decision Support*

2013-02: **Erietta Liarou** (CWI), *MonetDB/DataCell: Leveraging the Column-store Database Technology for Efficient and Scalable Stream Processing*

2013-03: **Szymon Klarman** (VU), *Reasoning with Contexts in Description Logics*

2013-04: **Chetan Yadati** (TUD), *Coordinating autonomous planning and scheduling*

2013-05: **Dulce Pumareja** (UT), *Groupware Requirements Evolutions Patterns*

2013-06: **Romulo Goncalves** (CWI), *The Data Cyclotron: Juggling Data and Queries for a Data Warehouse Audience*

2013-07: **Giel van Lankveld** (UvT), *Quantifying Individual Player Differences*

2013-08: **Robbert-Jan Merk** (VU), *Making enemies: cognitive modeling for opponent agents in fighter pilot simulators*

2013-09: **Fabio Gori** (RUN), *Metagenomic Data Analysis: Computational Methods and Applications*

2013-10: **Jeewanie Jayasinghe Arachchige** (UvT), *A Unified Modeling Framework for Service Design*

2013-11: **Evangelos Pournaras** (TUD), *Multi-level Reconfigurable Self-organization in Overlay Services*

2013-12: **Marian Razavian** (VU), *Knowledge-driven Migration to Services*

2013-13: **Mohammad Safiri** (UT), *Service Tailoring: User-centric creation of integrated IT-based homecare services to support independent living of elderly*

2013-14: **Jafar Tanha** (UVA), *Ensemble Approaches to Semi-Supervised Learning Learning*

2013-15: **Daniel Hennes** (UM), *Multiagent Learning - Dynamic Games and Applications*

2013-16: **Eric Kok** (UU), *Exploring the practical benefits of argumentation in multi-agent deliberation*

2013-17: **Koen Kok** (VU), *The PowerMatcher: Smart Coordination for the Smart Electricity Grid*

2013-18: **Jeroen Janssens** (UvT), *Outlier Selection and One-Class Classification*

2013-19: **Renze Steenhuizen** (TUD), *Coordinated Multi-Agent Planning and Scheduling*

2013-20: **Katja Hofmann** (UvA), *Fast and Reliable Online Learning to Rank for Information Retrieval*

2013-21: **Sander Wubben** (UvT), *Text-to-text generation by monolingual machine translation*

2013-22: **Tom Claassen** (RUN), *Causal Discovery and Logic*

2013-23: **Patricio de Alencar Silva** (UvT), *Value Activity Monitoring*

2013-24: **Haitham Bou Ammar** (UM), *Automated Transfer in Reinforcement Learning*

2013-25: **Agnieszka Anna Latoszek-Berendsen** (UM), *Intention-based Decision Support. A new way of representing and implementing clinical guidelines in a Decision Support System*

2013-26: **Alireza Zarghami** (UT), *Architectural Support for Dynamic Homecare Service Provisioning*

2013-27: **Mohammad Huq** (UT), *Inference-based Framework Managing Data Provenance*

2013-28: **Frans van der Sluis** (UT), *When Complexity becomes Interesting: An Inquiry into the Information eXperience*

2013-29: **Iwan de Kok** (UT), *Listening Heads*

2013-30: **Joyce Nakatumba** (TUE), *Resource-Aware Business Process Management: Analysis and Support*

2013-31: **Dinh Khoa Nguyen** (UvT), *Blueprint Model and Language for Engineering Cloud Applications*

2013-32: **Kamakshi Rajagopal** (OUN), *Networking For Learning; The role of Networking in a Lifelong Learner's Professional Development*

2013-33: **Qi Gao** (TUD), *User Modeling and Personalization in the Microblogging Sphere*

2013-34: **Kien Tjin-Kam-Jet** (UT), *Distributed Deep Web Search*

2013-35: **Abdallah El Ali** (UvA), *Minimal Mobile Human Computer Interaction*

2013-36: **Than Lam Hoang** (TUe), *Pattern Mining in Data Streams*

2013-37: **Dirk Brner** (OUN), *Ambient Learning Displays*

2013-38: **Eelco den Heijer** (VU), *Autonomous Evolutionary Art*

2013-39: **Joop de Jong** (TUD), *A Method for Enterprise Ontology based Design of Enterprise Information Systems*

2013-40: **Pim Nijssen** (UM), *Monte-Carlo Tree Search for Multi-Player Games*

2013-41: **Jochem Liem** (UVA), *Supporting the Conceptual Modelling of Dynamic Systems: A Knowledge Engineering Perspective on Qualitative Reasoning*

2013-42: **Lon Planken** (TUD), *Algorithms for Simple Temporal Reasoning*

2013-43: **Marc Bron** (UVA), *Exploration and Contextualization through Interaction and Concepts*

2014

2014-01: **Nicola Barile** (UU), *Studies in Learning Monotone Models from Data*

2014-02: **Fiona Tuliyano** (RUN), *Combining System Dynamics with a Domain Modeling Method*

2014-03: **Sergio Raul Duarte Torres** (UT), *Information Retrieval for Children: Search Behavior and Solutions*

2014-04: **Hanna Jochmann-Mannak** (UT), *Websites for children: search strategies and interface design - Three studies on children's search performance and evaluation*

2014-05: **Jurriaan van Reijsen** (UU), *Knowledge Perspectives on Advancing Dynamic Capability*

2014-06: **Damian Tamburri** (VU), *Supporting Networked Software Development*

2014-07: **Arya Adriansyah** (TUE), *Aligning Observed and Modeled Behavior*

2014-08: **Samur Araujo** (TUD), *Data Integration over Distributed and Heterogeneous Data Endpoints*

2014-09: **Philip Jackson** (UvT), *Toward Human-Level Artificial Intelligence: Representation and Computation of Meaning in Natural Language*

2014-10: **Ivan Salvador Razo Zapata** (VU), *Service Value Networks*

2014-11: **Janneke van der Zwaan** (TUD), *An Empathic Virtual Buddy for Social Support*

2014-12: **Willem van Willigen** (VU), *Look Ma, No Hands: Aspects of Autonomous Vehicle Control*

2014-13: **Arlette van Wissen** (VU), *Agent-Based Support for Behavior Change: Models and Applications in Health and Safety Domains*

2014-14: **Yangyang Shi** (TUD), *Language Models With Meta-information*
2014-15: **Natalya Mogles** (VU), *Agent-Based Analysis and Support of Human Functioning in Complex Socio-Technical Systems: Applications in Safety and Healthcare*
2014-16: **Krystyna Milian** (VU), *Supporting trial recruitment and design by automatically interpreting eligibility criteria*
2014-17: **Kathrin Dentler** (VU), *Computing healthcare quality indicators automatically: Secondary Use of Patient Data and Semantic Interoperability*
2014-18: **Mattijs Ghijsen** (VU), *Methods and Models for the Design and Study of Dynamic Agent Organizations*
2014-19: **Vincius Ramos** (TUE), *Adaptive Hypermedia Courses: Qualitative and Quantitative Evaluation and Tool Support*
2014-20: **Mena Habib** (UT), *Named Entity Extraction and Disambiguation for Informal Text: The Missing Link*
2014-21: **Kassidy Clark** (TUD), *Negotiation and Monitoring in Open Environments*
2014-22: **Marieke Peeters** (UU), *Personalized Educational Games - Developing agent-supported scenario-based training*
2014-23: **Eleftherios Sidirourgos** (UvA/CWI), *Space Efficient Indexes for the Big Data Era*
2014-24: **Davide Ceolin** (VU), *Trusting Semi-structured Web Data*
2014-25: **Martijn Lappenschaar** (RUN), *New network models for the analysis of disease interaction*
2014-26: **Tim Baarslag** (TUD), *What to Bid and When to Stop*
2014-27: **Rui Jorge Almeida** (EUR), *Conditional Density Models Integrating Fuzzy and Probabilistic Representations of Uncertainty*
2014-28: **Anna Chmielowiec** (VU), *Decentralized k-Clique Matching*
2014-29: **Jaap Kabbedijk** (UU), *Variability in Multi-Tenant Enterprise Software*
2014-30: **Peter de Cock** (UvT), *Anticipating Criminal Behaviour*
2014-31: **Leo van Moergestel** (UU), *Agent Technology in Agile Multiparallel Manufacturing and Product Support*
2014-32: **Naser Ayat** (UvA), *On Entity Resolution in Probabilistic Data*
2014-33: **Tesfa Tegegne** (RUN), *Service Discovery in eHealth*
2014-34: **Christina Manteli** (VU), *The Effect of Governance in Global Software Development: Analyzing Transactive Memory Systems*
2014-35: **Joost van Ooijen** (UU), *Cognitive Agents in Virtual Worlds: A Middleware Design Approach*
2014-36: **Joos Buijs** (TUE), *Flexible Evolutionary Algorithms for Mining Structured Process Models*
2014-37: **Maral Dadvar** (UT), *Experts and Machines United Against Cyberbullying*
2014-38: **Danny Plass-Oude Bos** (UT), *Making brain-computer interfaces better: improving usability through post-processing*
2014-39: **Jasmina Maric** (UvT), *Web Communities, Immigration, and Social Capital*
2014-40: **Walter Omona** (RUN), *A Framework for Knowledge Management Using ICT in Higher Education*
2014-41: **Frederic Hogenboom** (EUR), *Automated Detection of Financial Events in News Text*
2014-42: **Carsten Eijckhof** (CWI/TUD), *Contextual Multidimensional Relevance Models*
2014-43: **Kevin Vlaanderen** (UU), *Supporting Process Improvement using Method Increments*
2014-44: **Paulien Meesters** (UvT), *Intelligent Blauw. Met als ondertitel: Intelligence-gestuurde politiezorg in gebiedsgebonden eenheden*
2014-45: **Birgit Schmitz** (OUN), *Mobile Games for Learning: A Pattern-Based Approach*
2014-46: **Ke Tao** (TUD), *Social Web Data Analytics: Relevance, Redundancy, Diversity*
2014-47: **Shangsong Liang** (UVA), *Fusion and Diversification in Information Retrieval*

2015

2015-01: **iels Netten** (UVA), *Machine Learning for Relevance of Information in Crisis Response*
2015-02: **aiza Bukhsh** (UvT), *Smart auditing: Innovative Compliance Checking in Customs Controls*
2015-03: **wan van Laarhoven** (RUN), *Machine learning for network data*
2015-04: **oward Spoelstra** (OU), *Cooperations in Open Learning environments*
2015-05: **hristoph Bsch** (UT), *Cryptographically Enforced Search Pattern Hiding*

2015-06: **arideh Heidari** (TUD), *Business Process Quality Computation – Computing Non-Functional Requirements to Improve Business Processes*

2015-07: **aria-Hendrike Peetz** (UVA), *Time-Aware Online Reputation Analysis*

2015-08: **Jie Jiang** (TUD), *Organizational Compliance: An agent-based model for designing and evaluating organizational interactions*